SANDIA REPORT

SAND2005-7563J
Unlimited Release
Printed December 2005

Stabilized shock hydrodynamics: I. A Lagrangian method.

Guglielmo Scovazzi, Sandia
Mark Christon, ABAQUS, Inc.
Thomas Hughes, University of Texas at Austin
John Shadid, Sandia

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/ordering.htm

SAND2005-7563J
Unlimited Release
Printed December 2005

Stabilized shock hydrodynamics: I. A Lagrangian method.

Guglielmo Scovazzi
1431 Computational Shock- and Multi-physics Department
Sandia National Laboratories
P.O. Box 5800, MS 1110
Albuquerque, NM 87185-1110, USA
gscovaz@sandia.gov

Mark A. Christon
ABAQUS, Inc.
Rising Sun Mills, 166 Valley Street
Providence, RI 02909-2499, USA

Thomas J.R. Hughes
Institute for Computational Engineering and Sciences
The University of Texas at Austin
1 University Station C0200
Austin, TX 78712-0027, USA

John N. Shadid
1437 Computational Sciences R&D Group
Sandia National Laboratories
P.O. Box 5800, MS 0316
Albuquerque, NM 87185-0316, USA

Abstract

A new SUPG-stabilized formulation for Lagrangian hydrodynamics of materials satisfying Mie-Grüneisen equation of state. It allows the use of simplex-type (triangular/tetrahedral) meshes as well as the more commonly used brick-type (quadrilateral/hexahedral) meshes. The proposed method yields a globally conservative formulation, in which equal-order interpolation (P1 or Q1 isoparametric finite elements) is applied to velocities, displacements, and pressure. As a direct consequence, and in contrast to traditional cell-centered multidimensional hydrocode implementations, the proposed formulation allows a natural representation of the pressure gradient on element interiors. The SUPG stabilization involves additional design requirements, specific to the Lagrangian formulation. A discontinuity capturing operator in the form of a Noh-type viscosity with artificial heat flux is used to preserve stability and smoothness of the solution in shock regions. A set of challenging shock hydrodynamics benchmark tests for the Euler equations of gasdynamics in one and two space dimensions is presented. In the two-dimensional case, computations performed on quadrilateral and triangular grids are analyzed and compared. These results indicate that the new formulation is a promising technology for hydrocode applications.

Acknowledgments

This research was partially funded by the DOE NNSA Advanced Scientific Computing Program and the Computer Science Research Institute at Sandia National Laboratories.

We would like to thank Tim Trucano, at Sandia National Laboratories, for providing many very insightful comments on a preliminary draft of the manuscript, as well as helpful suggestions during the testing of the algorithm, Misha Shashkov at Los Alamos National Laboratories for comments and enriching discussions on various aspects of hydrocode technology, and Pavel Bochev and Rich Lehoucq for helpful discussions on mathematical aspects of the formulation. We would also like to thank Allen Robinson, Clark Dohrmann and Tom Voth for suggestions and help that improved aspects of the method.

G. Scovazzi would like to express special thanks to Randy Summers at Sandia National Laboratories for his genuine and cheerful encouragement. Thomas J. R. Hughes has been supported by Sandia National Laboratories contract No. A0340.0 with the University of Texas at Austin. This support is gratefully acknowledged.

Contents

1 Introduction 14

2 Equations of Lagrangian Hydrodynamics 16

2.1 Mie-Grüneisen constitutive laws 17

2.2 The one-dimensional hydrodynamics equations 18

2.2.1 Conservative form 18

2.2.2 Quasi-linear form 18

2.3 The two-dimensional hydrodynamics equations 20

2.3.1 Conservative form 20

2.3.2 Quasi-linear form 21

3 Time integration and variational equations 24

3.1 A second-order Petrov-Galerkin time integrator 24

3.2 Space-time variational formulation 26

3.2.1 Euler-Lagrange equations 28

3.3 Second-order in time formulation 28

4 SUPG Stabilization 31

4.1 A multiscale perspective 31

4.2 A new design requirement: Galilean invariance 33

4.3 Stabilization in the Lagrangian framework 34

5 Discontinuity capturing operator **36**

5.1 The mesh parameter “$h_{\boldsymbol{b}}$” 39

5.1.1 A limiting strategy for the mesh parameter 39

5.1.2 Quadrilaterals 40

5.1.3 Triangles 40

6 Global conservation **42**

7 Explicit predictor/multi-corrector algorithm **43**

7.1 The general nonlinear system 44

7.2 Newton solver 44

7.3 Assembly 45

7.3.1 Time-integration strategy for displacements 46

8 General considerations on implementation **48**

8.1 Post-processed variables 48

8.2 Proper specification of the density initial condition 49

8.3 Post-processing of the current configuration density 50

8.4 CFL condition 51

9 One-dimensional tests **54**

9.1 Sod’s problem 55

9.2 Left-half of Woodward-Colella blast 55

9.3 Two-shock problem 58

9.4 Planar Noh’s test 58

9.5 Overall assessment for the one-dimensional tests 61

10 Two-dimensional numerical tests **62**

10.1 Preliminary test on triangular meshes: Piston problem 62

10.2 Saltzmann test in cartesian coordinates 64
10.3 Two-dimensional Noh test on cartesian grids 74
    10.3.1 Triangular meshes on the unit circle 74
    10.3.2 Orthogonal grids on the unit quadrant 83
10.4 Sedov two-dimensional blast test 91

11 Summary **99**

Appendix

A An introduction to SUPG stabilization **101**

B One-dimensional stabilization [44] **104**

B.1 Second-order time integrator 105

References **110**

List of Figures

3.1 Second-order algorithm: Sketch of the typical solution (blue) and test function (red). 25

3.2 General finite element discretization in space-time. 27

3.3 Local trial functions for the second-order time integrator, in the one-dimensional case. ξ is the local space coordinate, while η is the local time coordinate. 29

3.4 Local test functions for the second-order time integrator, in the one-dimensional case. ξ is the local space coordinate, while η is the local time coordinate. 29

5.1 Sketch of the segments connecting the mid-points for quadrilaterals (left) and triangles (right). 40

8.1 Sketch of the preprocessing procedure for pressures and densities in one space dimension. 50

8.2 Behavior of $\Delta t/\Delta t_{adv}$ as a function of Pe_ν. The optimum is to have $\Delta t/\Delta t_{adv} = 1$, that is, the CFL condition is uniquely due to the speed of sound. This situation is achieved in the limit $Pe_\nu \to \infty$. 52

9.1 Hydrocode solution for the Sod test. The exact solution is represented by the continuous line. 56

9.2 SUPG solution for the Sod tests. The exact solution is represented by the continuous line. 56

9.3 Hydrocode solution for the left-half of the Woodward-Colella blast test. The exact solution is represented by the continuous line. 57

9.4 SUPG solution for the left-half of the Woodward-Colella blast test. The exact solution is represented by the continuous line. 57

9.5 Hydrocode solution for the two-shock problem. The exact solution is represented by the continuous line. 59

9.6 SUPG solution for the two-shock problem. The exact solution is represented by the continuous line. 59

9.7 Hydrocode solution for the one-dimensional, planar Noh test. The exact solution is represented by the continuous line. 60

9.8 Second-order Hydro-SUPG for the one-dimensional, planar Noh test. The exact solution is represented by the continuous line. 60

10.1 Piston problem for triangles: Initial meshes. 63

10.2 Piston problem for triangles: Pressure. The two solutions are evaluated at the same time step. Notice the absence of any node-to-node oscillations, typical of artificially stiff P1 elements. 63

10.3 Initial mesh for the Saltzmann 1:1 test. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). The 1:2 and 100:1 meshes are obtained by appropriate scaling along the horizontal axis. 65

10.4 1:1 Saltzmann solution at $T = 0.7$. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). 66

10.5 1:2 Saltzmann solution at $T = 0.7$. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). 67

10.6 100:1 Saltzmann solution at $T = 0.7$. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). The horizontal axis has been rescaled by a factor 0.01, for presentation purposes. 68

10.7 Saltzmann 1:1 test. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). Along the rows from top to bottom are plotted – as a function of the vertical coordinate y – pressure, normalized horizontal velocity ($v_1/\max(|v_2|)$), vertical velocity v_2, and nodal density. The exact solution is represented by the continuous red line, the dots represent *all* the nodal values of the numerical solution (the x-coordinate locations are compressed into a single plane). 69

10.8 Saltzmann 1:2 test. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). Along the rows from top to bottom are plotted – as a function of the vertical coordinate y – pressure, normalized horizontal velocity ($v_1/\max(|v_2|)$), vertical velocity v_2, and nodal density. The exact solution is represented by the continuous red line, the dots represent *all* the nodal values of the numerical solution (the x-coordinate locations are compressed into a single plane). 70

10.9 Saltzmann 100:1 test. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). Along the rows from top to bottom are plotted – as a function of the vertical coordinate y – pressure, normalized horizontal velocity ($v_1/\max(|v_2|)$), vertical velocity v_2, and nodal density. The exact solution is represented by the continuous red line, the dots represent *all* the nodal values of the numerical solution (the x-coordinate locations are compressed into a single plane). 71

10.10 Saltzmann 1:1 test solution (density color plot). Left: Quadrilaterals (mesh (I)) at $T = 0.925$. Center: Triangles (mesh (II)) at $T = 0.925$. Right: Triangles (mesh (III)) at $T \approx 0.85$. The white horizontal marker indicates the location of the shock for the exact solution. Notice that the 1:1 test on mesh (III) could not be run to the end time due to a volume inversion. 72

10.11 Saltzmann 100:1 test solution (density color plot) at $T = 0.925$. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). The white horizontal marker indicates the location of the shock for the exact solution. 72

10.12 Results from [8] on the Saltzmann test. 73

10.13 Initial configuration for mesh (a), 3,160 triangular elements, 1,601 nodes. 74

10.14 Initial configuration for mesh (b), 3,052 triangular elements, 1,541 nodes. 75

10.15 Zoom of the initial grid near the origin (left) and of the initial velocity (right). Top: Mesh (a). Center: Mesh (b). Bottom: Mesh (c), 12,376 triangular elements, 6,217 nodes. 76

10.16 Noh test on mesh (a): Final grid (above) and density color plot (below). 78

10.17 Noh test on mesh (b): Final grid (above) and density color plot (below). 79

10.18 Noh test on mesh (c): Final grid (above) and density color plot (below). 80

10.19 Noh test on a quadrant: Density elevation plots. Top: Mesh (a). Center: Mesh (b). Bottom: Mesh (c). 81

10.20 Noh test performance comparison. Left: Mesh (a). Center: Mesh (b). Right: Mesh (c). Along the rows from top to bottom are plotted – as a function of the radius $r = \sqrt{x^2 + y^2}$ – pressure, radial velocity v_r, normalized tangential velocity $(v_t/\max(|v_r|))$, and nodal density, respectively. The exact solution is represented by the continuous line, the dots represent the nodal values of the numerical solution. All plots contain information from each angularly-displaced radial line. 82

10.21 Initial meshes and velocity conditions for the Noh test on a quadrant. Top: Mesh (I). Center: Mesh (II). Bottom: Mesh (III). 84

10.22 Noh test on a quadrant, mesh (I). Top: Deformed grid. Bottom: Contour plot of the density. 85

10.23 Noh test on a quadrant, mesh (II). Top: Deformed grid. Bottom: Contour plot of the density. 86

10.24 Noh test on a quadrant, mesh (III). Top: Deformed grid. Bottom: Contour plot of the density. 87

10.25 Noh test: Density elevation plots. Top: Mesh (I). Center: Mesh (II). Bottom: Mesh (III). 88

10.26 Noh test performance comparison. Left: Mesh (I). Center: Mesh (II). Right: Mesh (III). Along the rows from top to bottom are plotted – as a function of the radius $r = \sqrt{x^2 + y^2}$ – pressure, radial velocity v_r, normalized tangential velocity $(v_t/\max(|v_r|))$, and nodal density, respectively. The exact solution is represented by the continuous line, the dots represent the nodal values of the numerical solution. All plots contain information from each angularly-displaced radial line. 89

10.27 Results from [8] on the Noh test on the quadrant. The mesh is the same as the quadrilateral mesh (I). Above: Mesh deformation. Center: isolines for the density. Bottom: Density radial plot. 90

10.28 Sedov test on mesh (I). Top: Deformed grid. Bottom: Density contour plot. 92

10.29 Sedov test on mesh (II). Top: Deformed grid. Bottom: Density contour plot. 93

10.30 Sedov test on mesh (III). Top: Deformed grid. Bottom: Density contour plot. 94

10.31 Sedov test: Density elevation plots. Top: Mesh (I). Center: Mesh (II). Bottom: Mesh (III). 95

10.32 Sedov test performance comparison. Left: Mesh (I). Center: Mesh (II). Right: Mesh (III). Along the rows from top to bottom are plotted – as a function of the radius $r = \sqrt{x^2 + y^2}$ – pressure, radial velocity v_r, normalized tangential velocity $(v_t/\max(|v_r|))$, and nodal density, respectively. The exact solution is represented by the continuous red line, the dots represent the nodal values of the numerical solution. All plots contain information from each angularly-displaced radial line. . . . 96

10.33 Sedov test on 1:2 aspect ratio anisotropic mesh. Top: Grid deformation. Bottom: Density contour plot. 97

10.34 Sedov test on the half plane, quadrialteral mesh. Top: Grid deformation. Bottom: Density contour plot. 98

Chapter 1

Introduction

Since their inception during the Manhattan Project at Los Alamos, algorithms for Lagrangian shock hydrodynamics computations (*hydrocodes* in short) have found widespread application to the transient analysis of a broad class of problems, from structural and fluid mechanics, to high energy density physics and astrophysics.

The original numerical formulation used in hydrocodes has been substantially preserved over time. In general, all kinematic variables (displacements, velocities, and accelerations) are node-centered, thermodynamic variables are cell-centered, and the computational meshes used are almost universally brick-type (quadrilaterals in two dimensions and hexahedra in three dimensions).

The main differences between the various hydrocode formulations mostly reside in the definition of the artificial viscosity operator used to capture shocks, the numerical representation of gradient operators for the thermodynamic quantities, and the methodology adopted to control hourglass instabilities.

Despite a number of attempts in the past, there are still no reliable and efficient formulations for simplex-type meshes (triangular in two dimensions and tetrahedral in three dimensions). The reason lies in the artificial stiffness that affects finite element formulations for which thermodynamic variables (specifically the pressure) are discretized as piecewise constants. This drawback is particularly evident in the incompressible limit (occurring, for example, for plastic flow), for which the well known *locking* phenomenon can occur (see, e.g., Hughes [18]).

Another problematic issue affecting current hydrocode technology (see Christon [10] for an extensive discussion) is the reconstruction of the gradients of thermodynamic variables – represented, by piecewise constant functions. Numerical solutions are extremely sensitive to the type of reconstruction used. In the context of *mimetic* finite-difference schemes, the work of Caramana *et al.* [9] and Campbell and Shashkov [8] carefully discusses how to improve discretizations.

The proposed new method *bypasses* completely the issue of pressure gradient representation, which is considered by the authors a key aspect for improving on the current state-of-the-art. The simple idea adopted herein is to derive a *globally conservative* formulation, in which pressure and kinematic variables are represented by means of globally-continuous, piecewise-linear shape functions (with corresponding *nodal* degrees-of-freedom) and to *stabilize* this formulation by means of the SUPG/multiscale framework of Hughes and coworkers [6, 25, 22, 44, 23, 24]. Convegence analysis of the method in the context of advection-diffusion systems of equations is given in [28, 21, 20, 29, 30, 46].

From the point of view of gradient representation, the advantage of the new approach is clear, since on element interiors all first derivatives are well-defined without the need to resort to special reconstruction techniques. This aspect is particularly evident in the numerical results on brick-type meshes, which do not show most of the usual pathologies common to standard hydrocode simulations.

It the case of two space dimensions, little difference is observed between the results of simulations performed on quadrilateral and triangular meshes. This fact is of particular importance, since, to the best of our knowledge, there are no examples of accurate and robust algorithms for Lagrangian hydrodynamics on simplex-type computational grids. In particular, the results on triangular meshes in two dimensions show the absence of element artificial stiffness, while the simulations on quadrilateral meshes proved free from hourglassing.

In summary, the absence of either hourglassing or locking, and the quality of the results, usually equal, and in some instances superior, to state-of-the-art algorithms currently documented in the literature, suggest a significant potential for complex-geometry, multi-physics applications.

The rest of the exposition is organized as follows: the basic equations of Lagrangian hydrodynamics are introduced in chapter 2. The variational formulation for Lagrangian hydrodynamics applications is established in chapter 3, while chapter 4 is devoted to the design of the stabilization terms. The artificial viscosity adopted is described in chapter 5. Section 6 discusses the global conservation properties embedded in the variational formulation, and chapter 7 summarizes the predictor/multi-corrector time-integration strategy. Section 8 contains a general discussion about the time step CFL constraints for the method, as well as pre- and post-processing issues. Results of the numerical tests are analyzed in chapter 9 and chapter 10 for one and two space dimensions, respectively. Conclusions and future research perspectives are summarized in chapter 11. Readers not familiar with the concept of SUPG stabilization are invited to read Appendix A before the main body of the paper, to understand the foundations of stabilized methods in the simple context of a linear advection-diffusion problem in one dimension.

Chapter 2

Equations of Lagrangian Hydrodynamics

We begin by defining the *deformation* $\boldsymbol{\varphi}$ as the transformation from a reference configuration $\boldsymbol{X}$ (the *initial configuration*, corresponding to the material reference frame) to the *current configuration*:

$$\begin{aligned} \boldsymbol{\varphi} : V\ (open)\ \subset \mathbb{R}^{n_d} \quad &\rightarrow \quad \mathbb{R}^{n_d} && (2.1) \\ \boldsymbol{X} \quad &\mapsto \quad \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t), \quad \forall \boldsymbol{X} \in V,\ t \geq 0, && (2.2) \end{aligned}$$

where n_d is the number of spatial dimensions. The boundary of V is denoted by S. The *deformation gradient* is defined as $\boldsymbol{F} = \textbf{Grad}\ \boldsymbol{\varphi}$, that is, $F_{iA} = \partial x_i / \partial X_A, \quad i, A \in \{1, \ldots, n_d\}$, and $J = \det \boldsymbol{F}$ is the *Jacobian determinant* of the transformation. (The summation convention for repeated indices is used throughout.) The equations of Lagrangian hydrodynamics, consisting of mass, momentum and energy balances can be formulated with respect to the reference configuration as follows:

$$\begin{aligned} \dot{\boldsymbol{u}} \quad &= \quad \boldsymbol{v} && (2.3) \\ \rho\ J \quad &= \quad \rho_0(\boldsymbol{X}) && (2.4) \\ \rho_0\ \dot{\boldsymbol{v}} \quad &= \quad \rho_0\ \boldsymbol{g} + \textbf{Div}\boldsymbol{P} && (2.5) \\ \rho_0\ \dot{E} \quad &= \quad \rho_0\ \boldsymbol{g} \cdot \boldsymbol{v} + \rho_0\ r + \textbf{Div}(\boldsymbol{P}^T \boldsymbol{v} + \boldsymbol{Q}) && (2.6) \end{aligned}$$

Here, $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X}$ is the displacement vector, ρ_0 is the reference (initial) density, ρ is the current density, $\boldsymbol{v}$ is the velocity, $\boldsymbol{g}$ is the body force, $\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-T} = \boldsymbol{\sigma}\ \textbf{cof}\boldsymbol{F}$ is the first Piola stress tensor ($\boldsymbol{\sigma}$ is the Cauchy stress tensor in the current configuration), $E = e + \boldsymbol{v} \cdot \boldsymbol{v}/2$ is the total energy, the sum of the internal energy e and the kinetic energy $\boldsymbol{v} \cdot \boldsymbol{v}/2$, r is the energy source term, and $\boldsymbol{Q} = J\boldsymbol{F}^{-1}\boldsymbol{q} = (\textbf{cof}\boldsymbol{F})^T\boldsymbol{q}$ is the Piola-transformed heat flux. E, e, $\boldsymbol{g}$, r are measured per unit mass. The reader interested in the derivations of the Lagrangian equations in the reference configuration may refer to [41], [2], or [38] for detailed derivations.

For the sake of simplicity, in the discussion that follows, body forces, heat transfer, and internal sources of energy are assumed absent. Also, the so-called Piola identity

$$\mathbf{Div}\ \mathbf{cof}\boldsymbol{F} \equiv \mathbf{0} \tag{2.7}$$

will be extensively used in the subsequent derivations.

2.1 Mie-Grüneisen constitutive laws

The following analysis is specific to materials satisfying the Mie-Grüneisen equations of state, including ideal gases, co-volume gases, high explosives, and elastic-plastic solids with no strength (a situation that can be achieved when bulk stresses in the material are larger than shear stresses by orders of magnitude). No deformation strength is involved, so that the stress tensor $\boldsymbol{\sigma}$ reduces to an isotropic tensor, dependent only on the thermodynamic pressure:

$$\boldsymbol{\sigma} \;=\; -p\boldsymbol{I}_{n_d \times n_d} \tag{2.8}$$

or, in index notation,

$$\sigma_{ij} \;=\; -p\ \delta_{ij} \tag{2.9}$$

with δ_{ij}, the Kronecker tensor. Mie-Grüneisen materials satisfy an equation of state of the form $p = f_1(\rho; \rho_r, e_r) + f_2(\rho; \rho_r, e_r)e$, where ρ_r and e_r are fixed reference thermodynamic states. More succinctly,

$$p \;=\; f_1(\rho) \;+\; f_2(\rho)\ e \tag{2.10}$$

If $f_1 = 0$ and $f_2 = (\gamma - 1)\ \rho$, the equation of state for an ideal gas, $p \;=\; (\gamma - 1)\ \rho\ e$, is recovered. It will prove very useful, for the derivations in the next sections, to recast (2.10) as follows

$$e \;=\; g_1(\rho) \;+\; g_2(\rho)\ p \tag{2.11}$$

where $g_1 = -f_1/f_2$ and $g_2 = 1/f_2$. In addition

$$g_1' \;=\; \frac{dg_1}{d\rho} = -\frac{f_1' f_2 - f_1 f_2'}{f_2^2} = -\frac{f_1'}{f_2} + \frac{f_1 f_2'}{f_2^2} \tag{2.12}$$

$$g_2' \;=\; \frac{dg_2}{d\rho} = -\frac{f_2'}{f_2^2} \tag{2.13}$$

For ideal gases, it can be easily shown that $g_1 = 0$, $g_2 = 1/((\gamma - 1)\rho)$, $g_1' = 0$, and $g_2' = -1/((\gamma - 1)\rho^2)$. The numerical examples will be focused on ideal gases, but it is important to realize that the framework developed includes a broader class of materials.

2.2 The one-dimensional hydrodynamics equations

In the one dimensional case, $\boldsymbol{X}$ is a scalar, so that:

$$\begin{aligned}
\boldsymbol{X} &\equiv X_1 = X && (2.14)\\
\boldsymbol{x} &\equiv x_1 = x = \varphi(X,t) && (2.15)\\
\boldsymbol{u} &\equiv \boldsymbol{x} - \boldsymbol{X} = \varphi(X,t) - X = u && (2.16)\\
\boldsymbol{F} &\equiv F = \frac{\partial \varphi}{\partial X} = 1 + \frac{\partial u}{\partial X} && (2.17)\\
J &\equiv \det(\boldsymbol{F}) = \frac{\partial \varphi}{\partial X} \equiv F && (2.18)
\end{aligned}$$

In particular, (2.18) leads to the following simplification for the Piola stress tensor:

$$\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-T} = J\sigma J^{-1} = -p \tag{2.19}$$

The complete set of one-dimensional equations is given by

$$\begin{aligned}
\dot{u} &= v && (2.20)\\
\rho_0\ \dot{v} &= -\frac{\partial p}{\partial X} && (2.21)\\
\rho_0\ \dot{E} &= -\frac{\partial}{\partial X}(vp) && (2.22)\\
\rho\ J &= \rho_0 && (2.23)
\end{aligned}$$

2.2.1 Conservative form

Equations (2.20-2.22) can be cast into conservative form:

$$\dot{\boldsymbol{U}} + \boldsymbol{G}_{1,1} + \boldsymbol{Z} = \boldsymbol{0} \tag{2.24}$$

where $\boldsymbol{G}_{1,1} = \dfrac{\partial \boldsymbol{G}_1}{\partial X_1}$. In particular,

$$\boldsymbol{U} = \begin{bmatrix} u \\ \rho_0 v \\ \rho_0 E \end{bmatrix} \qquad \boldsymbol{G}_1 = \begin{bmatrix} 0 \\ p \\ pv \end{bmatrix} \qquad \boldsymbol{Z} = \begin{bmatrix} -v \\ 0 \\ 0 \end{bmatrix} \tag{2.25}$$

2.2.2 Quasi-linear form

In order to develop the SUPG stabilization operator it is essential to rearrange equations (2.20-2.22) in quasi-linear form, as follows:

$$\boldsymbol{A}_0\dot{\boldsymbol{Y}} + \boldsymbol{A}_1\boldsymbol{Y}_{,1} + \boldsymbol{C}\boldsymbol{Y} = \boldsymbol{0} \tag{2.26}$$

where

$$\boldsymbol{A}_0 = \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{Y}}, \qquad \boldsymbol{A}_1 = \frac{\partial \boldsymbol{G}_1}{\partial \boldsymbol{Y}}, \qquad \boldsymbol{C} = \frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{Y}} \tag{2.27}$$

are the Jacobian matrices for the temporal and spatial fluxes, respectively. We still have to specify $\boldsymbol{Y}$. It becomes clear that if continuous piecewise-linear functions for the kinematic and at least one of the thermodynamic variables are to be used, then the pressure is the natural candidate in the definition of Y. In fact, alternative options, such as *conservation variables*, create the problem of taking derivatives of J with respect to X when computing the matrix $\boldsymbol{A}_1$. The derivative of J with respect to X involves second derivatives of the displacements u, represented by Dirac distributions over element edges if piecewise linear interpolation is used, with possible adverse consequences in the design of the SUPG stabilization operator. Therefore,

$$\boldsymbol{Y} = \begin{bmatrix} u \\ v \\ p \end{bmatrix} \tag{2.28}$$

The following algebraic manipulations apply to $\dot{E}$:

$$\begin{aligned}
\rho_0 \dot{E} &= \rho_0(\dot{e} + v\dot{v}) \\
&= \rho_0 \, (\, (g_1' + g_2' \, p)\dot{\rho} \; + \; g_2 \dot{p} \,) + \rho_0 v \dot{v} \qquad (\text{ using } (2.11)\text{–}(2.13) \,) \\
&= -(g_1' + g_2' \, p)\frac{\rho_0^2}{J^2}\dot{J} + \rho_0 \, g_2 \dot{p} + \rho_0 v \dot{v} \qquad (\text{ using } (2.4) \,) \\
&= \rho_0 \, g_2 \dot{p} - (g_1' + g_2' \, p)\frac{\rho_0^2}{J^2}\frac{\partial v}{\partial X} + \rho_0 v \dot{v}
\end{aligned} \tag{2.29}$$

where we have used the fact that $\dot{J} = \frac{\partial \dot{u}}{\partial X} = \frac{\partial v}{\partial X}$. Notice also that the term involving $\frac{\partial v}{\partial X}$ does not contain a temporal derivative, and is therefore incorporated in the definition of the Jacobian $\boldsymbol{A}_1$. This final manipulation yields:

$$\boldsymbol{A}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \rho_0 & 0 \\ 0 & 0 & \rho_0 \; g_2\big(\frac{\rho_0}{J}\big) \end{bmatrix}, \quad \boldsymbol{A}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & \Psi\big(\frac{\rho_0}{J}, p\big) & 0 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.30}$$

where, using (2.4),

$$\Psi\Big(\frac{\rho_0}{J}, p\Big) = p \; - \; \Big(\frac{\rho_0}{J}\Big)^2 \Big(\; g_1'\Big(\frac{\rho_0}{J}\Big) + g_2'\Big(\frac{\rho_0}{J}\Big) \;\; p \; \Big) \tag{2.31}$$

In the case of an ideal gas,

$$\rho_0 \dot{E} \;\; = \;\; \frac{J}{\gamma - 1} \, \dot{p} + \frac{p}{\gamma - 1}\frac{\partial v}{\partial X} + \rho_0 v \dot{v} \tag{2.32}$$

so that

$$\boldsymbol{A}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \rho_0 & 0 \\ 0 & 0 & \frac{J}{\gamma-1} \end{bmatrix}, \qquad \boldsymbol{A}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & \frac{\gamma}{\gamma-1}p & 0 \end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.33}$$

Remark 1 *Notice that the Jacobians $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$ have been obtained using a crucial simplification: The term given by the inner product of the momentum equation times the velocity, the so-called kinetic energy equation, has been removed from the total energy equation. This amounts to writing a quasilinear equation for the internal energy rather than the total energy. With this choice, when the quasi-linear form of the Lagrangian hydrodynamics equations is used in the stabilization, the resulting SUPG term is invariant under Galilean transformations. Failure to respect Galilean invariance resulted in numerical instabilities for early tests, as further discussed in [40].*

2.3 The two-dimensional hydrodynamics equations

In the two-dimensional case, the manipulations are slightly more involved. The following notation is adopted:

$$\boldsymbol{X} \equiv \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{2.34}$$

$$\boldsymbol{x} \equiv \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \varphi_1(X_1, X_2, t) \\ \varphi_2(X_1, X_2, t) \end{bmatrix} \tag{2.35}$$

$$\boldsymbol{u} \equiv \boldsymbol{x} - \boldsymbol{X} = \begin{bmatrix} \varphi_1(X_1, X_2, t) \\ \varphi_2(X_1, X_2, t) \end{bmatrix} - \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.36}$$

$$\boldsymbol{F} \equiv \frac{\partial \varphi}{\partial \boldsymbol{X}} = \boldsymbol{I}_{2\times 2} + \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}} = \begin{bmatrix} 1 + u_{1,1} & u_{1,2} \\ u_{2,1} & 1 + u_{2,2} \end{bmatrix} \tag{2.37}$$

$$J \equiv \det(\boldsymbol{F}) = (1 + u_{1,1})(1 + u_{2,2}) - u_{2,1}u_{1,2} \tag{2.38}$$

where $u_{i,A} = \partial u_i / \partial X_A$. In particular, (2.8) and (2.37) lead to the following expression for the Piola stress tensor:

$$\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-T} = -p \begin{bmatrix} 1 + u_{2,2} & -u_{2,1} \\ -u_{1,2} & 1 + u_{1,1} \end{bmatrix} \tag{2.39}$$

2.3.1 Conservative form

The two-dimensional conservative form reads:

$$\dot{\boldsymbol{U}}(\boldsymbol{Y}) + \boldsymbol{G}_{i,i}(\boldsymbol{Y}) + \boldsymbol{Z}(\boldsymbol{Y}) = \boldsymbol{0} \tag{2.40}$$

where $\boldsymbol{G}_{i,A} = \partial \boldsymbol{G}_i / \partial X_A$, $i, A = 1, \ldots, n_d = 2$, are the derivatives of the spatial fluxes $\boldsymbol{G}_i(\boldsymbol{Y})$. Namely:

$$\boldsymbol{Y} = \begin{bmatrix} u_1 \\ u_2 \\ v_1 \\ v_2 \\ p \end{bmatrix}, \qquad \boldsymbol{U} = \begin{bmatrix} u_1 \\ u_2 \\ \rho_0 v_1 \\ \rho_0 v_2 \\ \rho_0 E \end{bmatrix}, \qquad \boldsymbol{Z} = \begin{bmatrix} -v_1 \\ -v_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.41}$$

$$\boldsymbol{G}_1 = \begin{bmatrix} 0 \\ 0 \\ p(1+u_{2,2}) \\ -pu_{1,2} \\ pv_1(1+u_{2,2}) - pv_2 u_{1,2} \end{bmatrix} \qquad \boldsymbol{G}_2 = \begin{bmatrix} 0 \\ 0 \\ -pu_{2,1} \\ p(1+u_{1,1}) \\ -pv_1 u_{2,1} + pv_2(1+u_{1,1}) \end{bmatrix} \tag{2.42}$$

2.3.2 Quasi-linear form

In order to design the SUPG stabilization, equations (2.40) need to be cast in quasi-linear form:

$$\boldsymbol{A}_0 \dot{\boldsymbol{Y}} + \boldsymbol{A}_1 \boldsymbol{Y}_{,1} + \boldsymbol{A}_2 \boldsymbol{Y}_{,2} + \boldsymbol{C}\boldsymbol{Y} = \boldsymbol{0} \tag{2.43}$$

where

$$\boldsymbol{A}_0 = \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{Y}}, \qquad \boldsymbol{A}_1 = \frac{\partial \boldsymbol{G}_1}{\partial \boldsymbol{Y}}, \qquad \boldsymbol{A}_2 = \frac{\partial \boldsymbol{G}_2}{\partial \boldsymbol{Y}}, \qquad \boldsymbol{C} = \frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{Y}} \tag{2.44}$$

are the Jacobian matrices. Note that $\boldsymbol{Z}$ is a linear function of $\boldsymbol{Y}$. Therefore, $\boldsymbol{Z}(\boldsymbol{Y}) = \boldsymbol{C}\boldsymbol{Y}$, where $\boldsymbol{C}$ is a constant matrix, justifying (2.43). The following definition of $\boldsymbol{Y}$ is used:

$$\boldsymbol{Y} = [u_1 \; u_2 \; v_1 \; v_2 \; p]^T \tag{2.45}$$

This choice prompts the following manipulations in the energy equation:

$$\begin{aligned} \rho_0 \dot{E} &= \rho_0(\dot{e} + v_1 \dot{v}_1 + v_2 \dot{v}_2) \\ &= \rho_0 \left(\; (g_1' + g_2' \; p)\dot{\rho} \; + \; g_2 \dot{p} \; \right) + \rho_0 v_1 \dot{v}_1 + \rho_0 v_2 \dot{v}_2 \\ &= \rho_0 \; g_2 \; \dot{p} - (g_1' + g_2' \; p)\frac{\rho_0^2}{J^2}\dot{J} + \rho_0 v_1 \dot{v}_1 + \rho_0 v_2 \dot{v}_2 \end{aligned} \tag{2.46}$$

with

$$\begin{aligned} \dot{J} &= \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{X}} ((1+u_{1,1})(1+u_{2,2}) - u_{2,1}u_{1,2}) \\ &= \quad (1+u_{2,2})v_{1,1} - u_{1,2}v_{2,1} \qquad \text{(term contributing to } \boldsymbol{A}_1\text{)} \\ &\quad -u_{2,1}v_{1,2} + (1+u_{1,1})v_{2,2} \qquad \text{(term contributing to } \boldsymbol{A}_2\text{)} \end{aligned} \tag{2.47}$$

It is important now to notice a very important cancellation in the second-derivative terms relative to the momentum and energy equations in (2.43), due to (2.7). In fact, we can clearly see that:

$$\begin{aligned}
\boldsymbol{G}_{1,1}+\boldsymbol{G}_{2,2} &= \begin{bmatrix} 0 \\ 0 \\ p_{,1}(1+u_{2,2})+pu_{2,21}-p_{,2}u_{2,1}-pu_{2,12} \\ -p_{,1}u_{1,2}-pu_{1,21}+p_{,2}(1+u_{1,1})+pu_{1,12} \\ (\boldsymbol{G}_{1,1}+\boldsymbol{G}_{2,2})_5 \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \\ p_{,1}(1+u_{2,2})-p_{,2}u_{2,1} \\ -p_{,1}u_{1,2}+p_{,2}(1+u_{1,1}) \\ (\boldsymbol{G}_{1,1}+\boldsymbol{G}_{2,2})_5 \end{bmatrix}
\end{aligned} \tag{2.48}$$

due to commutativity of second mixed derivatives, and

$$\begin{aligned}
(\boldsymbol{G}_{1,1}+\boldsymbol{G}_{2,2})_5 &= p_{,1}v_1(1+u_{2,2})+pv_{1,1}(1+u_{2,2})+pv_1u_{2,21} \\
&\quad -p_{,1}v_2u_{1,2}-pv_{2,1}u_{1,2}-pv_2u_{1,21} \\
&\quad -p_{,2}v_1u_{2,1}-pv_{1,2}u_{2,1}-pv_1u_{2,12} \\
&\quad +p_{,2}v_2(1+u_{1,1})+pv_{2,2}(1+u_{1,1})+pv_2u_{1,12} \\
&= (v_1(1+u_{2,2})-v_2u_{1,2})p_{,1} \\
&\quad +(-v_1u_{2,1}+v_2(1+u_{1,1}))p_{,2} \\
&\quad +(1+u_{2,2})pv_{1,1}-u_{1,2}pv_{2,1} \\
&\quad -u_{2,1}pv_{1,2}+(1+u_{1,1})pv_{2,2}
\end{aligned} \tag{2.49}$$

Arrangement of the terms according to the structure of (2.43), yields:

$$\boldsymbol{A}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \rho_0 & 0 & 0 \\ 0 & 0 & 0 & \rho_0 & 0 \\ 0 & 0 & 0 & 0 & \rho_0 g_2\big(\frac{\rho_0}{J},p\big) \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.50}$$

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+u_{2,2} \\ 0 & 0 & 0 & 0 & -u_{1,2} \\ 0 & 0 & (1+u_{2,2})\Psi\big(\frac{\rho_0}{J},p\big) & -u_{1,2}\Psi\big(\frac{\rho_0}{J},p\big) & 0 \end{bmatrix} \tag{2.51}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -u_{2,1} \\ 0 & 0 & 0 & 0 & 1+u_{1,1} \\ 0 & 0 & -u_{2,1}\Psi\big(\frac{\rho_0}{J},p\big) & (1+u_{1,1})\Psi\big(\frac{\rho_0}{J},p\big) & 0 \end{bmatrix} \tag{2.52}$$

For an ideal gas, $\Psi\big(\frac{\rho_0}{J}, p\big) = \frac{\gamma}{\gamma-1}\, p$, so that:

$$\boldsymbol{A}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \rho_0 & 0 & 0 \\ 0 & 0 & 0 & \rho_0 & 0 \\ 0 & 0 & 0 & 0 & \frac{J}{\gamma-1} \end{bmatrix} \tag{2.53}$$

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+u_{2,2} \\ 0 & 0 & 0 & 0 & -u_{1,2} \\ 0 & 0 & \frac{\gamma}{\gamma-1}\, p\, (1+u_{2,2}) & -\frac{\gamma}{\gamma-1}\, p\, u_{1,2} & 0 \end{bmatrix} \tag{2.54}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -u_{2,1} \\ 0 & 0 & 0 & 0 & 1+u_{1,1} \\ 0 & 0 & -\frac{\gamma}{\gamma-1}\, p\, u_{2,1} & \frac{\gamma}{\gamma-1}\, p\, (1+u_{1,1}) & 0 \end{bmatrix} \tag{2.55}$$

The generalization to the three-dimensional case is straightforward and not reported here for the sake of brevity.

Remark 2 *Analogous to the one-dimensional case, in order to generate a stabilization term invariant under Galilean transformations, $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, and $\boldsymbol{A}_2$ have been obtained by removing the kinetic energy equation from the total energy equation.*

Chapter 3

Time integration and variational equations

The basis of the proposed method is a space-time variational formulation, to which the SUPG stabilization is intimately connected. We adopt a Petrov-Galerkin formulation in both space and time, resulting in a second-order accurate method in space and time. This time-integration strategy simplifies to a standard mid-point rule when a single-point quadrature rule in time is applied. For obvious performance reasons, *all* numerical tests presented herein were performed using this variant.

The time-integration scheme dates back to Hulme [26], Jamet [27], and Aziz and Monk [1]. More recently, Estep and French [12], French [13, 14], French and Jensen [15], and French and Paterson [16] present an extensive survey of previous work in the context of parabolic/second-order hyperbolic problems, and novel developments in the context of global error analysis, and adaptive time-step control.

3.1 A second-order Petrov-Galerkin time integrator

In order to clarify the nature of the numerical discretization, the basic time-integration algorithm is described for the case of an ordinary differential equation. Let us consider the initial-value problem:

$$\dot{y} = f(y(t)) \tag{3.1}$$

$$y(0) = y_0 \tag{3.2}$$

The general space-time Galerkin formulation for (3.1)–(3.2) over the interval $[t_n, t_{n+1}]$

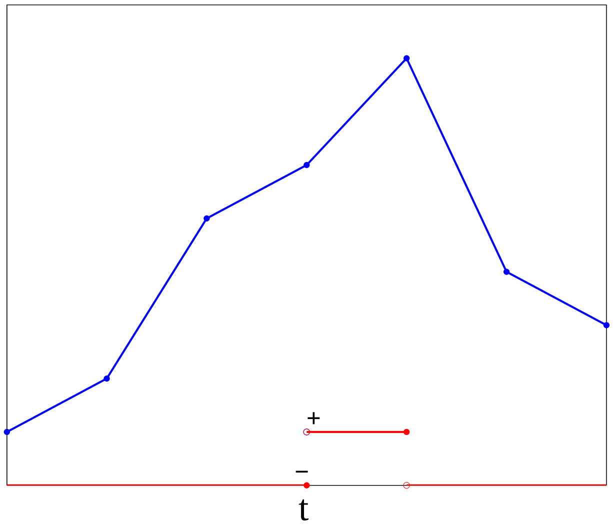

Figure 3.1. Second-order algorithm: Sketch of the typical solution (blue) and test function (red).

is given by:

$$
\begin{aligned}
0 &= w(t_{n+1}^{-})y(t_{n+1}^{-}) - w(t_n^{+})y(t_n^{-}) \\
&\quad - \int_{t_n}^{t_{n+1}} (\ w_{,s}(s)y(s) + w(s)f(y(s))\ )\ ds
\end{aligned}
\tag{3.3}
$$

Here $t_j^{\pm} = lim_{\varepsilon \to 0^{\pm}}(t_j + \varepsilon)$. Notice that, using integration by parts, the Euler-Lagrange equations for (3.3) over the time interval $[t_n, t_{n+1}]$ correspond exactly to (3.1) with (3.2) imposed *weakly*. Starting from the variational statement (3.3), a specific numerical scheme is defined by the function spaces representing the solution y and the test function w. A second-order accurate integrator is obtained by way of a Petrov-Galerkin formulation in time, for which the test space is given by discontinuous, piecewise-constant functions, and the trial space is represented by continuous, piecewise-linear functions, as shown in Figure 3.1. Then (3.3) reduces to:

$$
w^h(t_{n+1}^{-})y^h(t_{n+1}) - w^h(t_n^{+})y^h(t_n) - \int_{t_n}^{t_{n+1}} w^h(s)f(y^h(s))\ ds = 0 \tag{3.4}
$$

where w^h and y^h indicate the approximations of w and y.

Remark 3 *The current formulation requires only one update, namely* $y(t_{n+1})$.

Remark 4 *For linear systems (e.g., $f(y) = ay$), equation (3.4) reduces to the mid-point time-integration rule for the nodal degrees-of-freedom $y_k = y(t_k)$*

$$y_{n+1} - y_n = \frac{a\Delta t_n}{2}\left(y_{n+1} + y_n\right) \tag{3.5}$$

and to the well-known Crank-Nicolson scheme, in the case of a linear partial differential equation (PDE).

Remark 5 *For nonlinear equations, a mid-point scheme is also recovered when a single-point quadrature rule in time is applied. Such an approach was used in the numerical implementation of the SUPG method for Lagrangian hydrodynamics, since the algorithmic structure of the space-time formulation reduces to a classical semi-discrete time-integration scheme, with all the consequent advantages from a computational performance point of view.*

3.2 Space-time variational formulation

The variational formulation adopted in the Lagrangian hydrodynamics computations is now described. Given a partition $0 < t_1 < t_2 < \ldots < t_N = T$ of the time interval $I =]0, T]$, let $I_n =]t_n, t_{n+1}]$, so that $]0, T] = \bigcup_{n=0}^{N-1} I_n$. The space-time domain $Q = V \times I$ can be divided into time slabs

$$Q_n = V \times I_n \tag{3.6}$$

with "lateral" boundary $P_n = S \times I_n$. A sketch of the general discretization in space-time is presented in Figure 3.2. In general, the elements can assume fairly complex shapes in space-time. However, we will only make use of discretizations *prismatic* in time. The material domain V is further divided into material-subdomains V^e (elements in space, a partition of the initial configuration, fixed with respect to time). Thus $V = \overline{\bigcup_{e=1}^{n_{el}} V^e}$, and, consequently, a typical space-time element is given by the prism (i.e., tensor product domain)

$$Q_n^e = V^e \times I_n \tag{3.7}$$

It is also assumed that the space-time boundary is partitioned as $P_n = P_n^g \cup P_n^h$, $P_n^g \cap P_n^h = \emptyset$ (i.e., P_n is divided into a Dirichlet boundary P_n^g and a Neumann boundary P_n^h). Using the notation $\boldsymbol{V}(\boldsymbol{X}, t_n^{\pm}) = \lim_{t \to t_n^{\pm}} \boldsymbol{V}(\boldsymbol{X}, t)$, the classical space-time variational formulation is defined as follows:

Find $\boldsymbol{Y}^h \in \mathcal{S}^h$, such that $\forall \boldsymbol{W}^h \in \mathcal{V}^h$

$$\mathcal{B}(\boldsymbol{W}^h, \boldsymbol{Y}^h) + \mathcal{SUPG}(\boldsymbol{W}^h, \boldsymbol{Y}^h) + \mathcal{DC}(\boldsymbol{W}^h, \boldsymbol{Y}^h) \;=\; \mathcal{F}(\boldsymbol{W}^h) \tag{3.8}$$

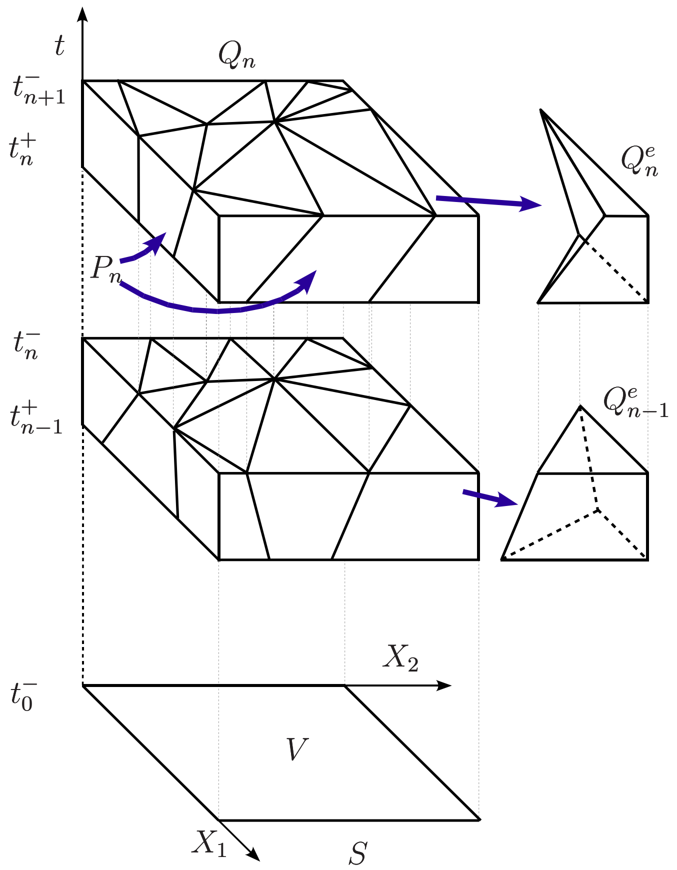

Figure 3.2. General finite element discretization in space-time.

where $\mathcal{B}(\cdot,\cdot)$ is the space-time bilinear form, $\mathcal{SUPG}(\cdot,\cdot)$ is the stabilization operator, and $\mathcal{DC}(\cdot,\cdot)$ is the discontinuity-capturing operator. $\mathcal{SUPG}(\cdot,\cdot)$ and $\mathcal{DC}(\cdot,\cdot)$ are linear in the first argument and nonlinear in the second. Specifically:

$$\begin{aligned}\mathcal{B}(\boldsymbol{W}^h,\boldsymbol{Y}^h) \quad = \quad & \int_V \boldsymbol{W}^h(\boldsymbol{X},t^-_{n+1})\cdot\boldsymbol{U}(\boldsymbol{Y}^h(\boldsymbol{X},t^-_{n+1}))dV \\ & -\int_V \boldsymbol{W}^h(\boldsymbol{X},t^+_n)\cdot\boldsymbol{U}(\boldsymbol{Y}^h(\boldsymbol{X},t^-_n))dV \\ & +\int_{Q_n}\left(-\boldsymbol{W}^h_{,t}\cdot\boldsymbol{U}(\boldsymbol{Y}^h)-\boldsymbol{W}^h_{,i}\cdot\boldsymbol{G}_i(\boldsymbol{Y}^h)+\boldsymbol{W}^h\cdot\boldsymbol{Z}(\boldsymbol{Y}^h)\right)dQ \\ & +\int_{P^g_n}\boldsymbol{W}^h\cdot\boldsymbol{G}_i(\boldsymbol{Y}^h)N_i\, dP \qquad (3.9)\end{aligned}$$

$$\mathcal{F}(\boldsymbol{W}^h) \;=\; -\int_{P_n^h} \boldsymbol{W}^h \cdot \boldsymbol{H}_i N_i \; dP \tag{3.10}$$

where $\boldsymbol{H}_i$ represents the Neumann flux along the i-th coordinate (a *traction*-type boundary condition in Lagrangian hydrodynamics). The choice of the spaces $\mathcal{S}^h$ and $\mathcal{V}^h$ defines the time integrator, and, in our case, causes (3.9) to further simplify.

3.2.1 Euler-Lagrange equations

The space-time formulation is best understood through the Euler-Lagrange equations, obtained by integration by parts, assuming sufficient regularity of the solution:

$$\begin{aligned} &\int_{Q_n} \boldsymbol{W}^h \cdot \left\{ \boldsymbol{U}_{,t}(\boldsymbol{Y}^h) + \boldsymbol{G}_{i,i}(\boldsymbol{Y}^h) + \boldsymbol{Z}(\boldsymbol{Y}^h) \right\} \; dQ \\ &+ \int_V \boldsymbol{W}^h(\boldsymbol{X}, t_n^+) \cdot \left\{ \boldsymbol{U}(\boldsymbol{Y}^h(\boldsymbol{X}, t_n^+)) - \boldsymbol{U}(\boldsymbol{Y}^h(\boldsymbol{X}, t_n^-)) \right\} \; dV \\ &- \int_{P_n^h} \boldsymbol{W}^h \cdot \left\{ \boldsymbol{G}_i(\boldsymbol{Y}^h) - \boldsymbol{H}_i \right\} N_i dP \\ &+ \mathcal{SUPG}(\boldsymbol{W}^h, \boldsymbol{Y}^h) + \mathcal{DC}(\boldsymbol{W}^h, \boldsymbol{Y}^h) \;=\; 0 \end{aligned} \tag{3.11}$$

The integral over the space-time domain Q_n in (3.11) tests the system of PDEs on the space-time domain, the integral over the spatial domain V enforces weak continuity of the solution across time-slabs, according to the *causality* principle of temporal evolution. The integral over the lateral space-time surface P_n^h on the third line tests the Neumann boundary conditions (Dirichlet conditions are embedded in the definition of the trial functions), while the last two terms are yet to be defined.

3.3 Second-order in time formulation

In terms of function spaces, we will assume that the trial function space $\mathcal{S}_n$ is given by the piecewise-linear, continuous functions on $Q = V \times]0, T[$ (see Fig. 3.3), while the test function space $\mathcal{V}^h$ will be given by functions that are continuous piecewise-linear in space and discontinuous, piecewise-constant in time (see Fig. 3.4 for details in the one-dimensional case). Therefore,

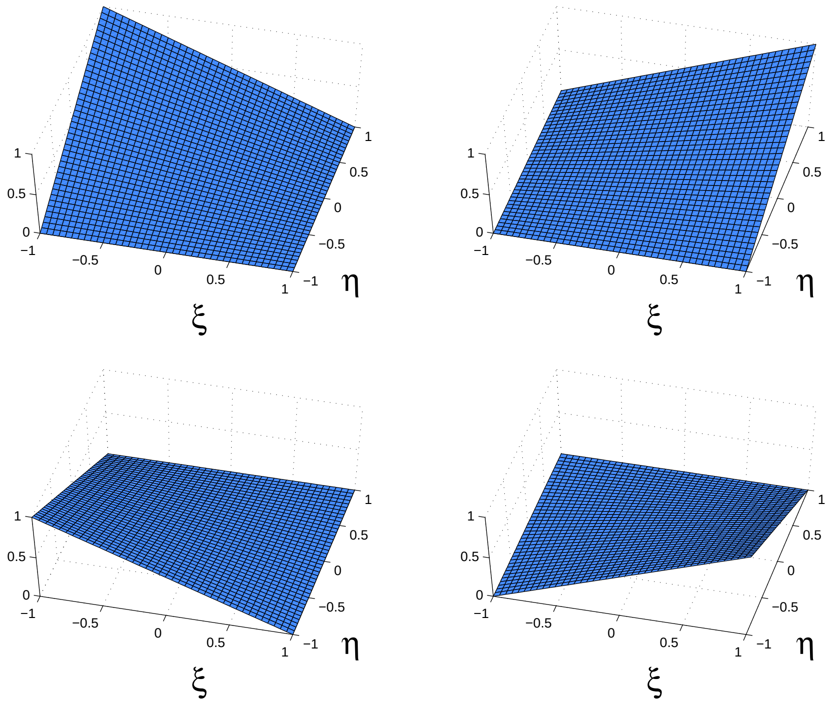

Figure 3.3. Local trial functions for the second-order time integrator, in the one-dimensional case. ξ is the local space coordinate, while η is the local time coordinate.

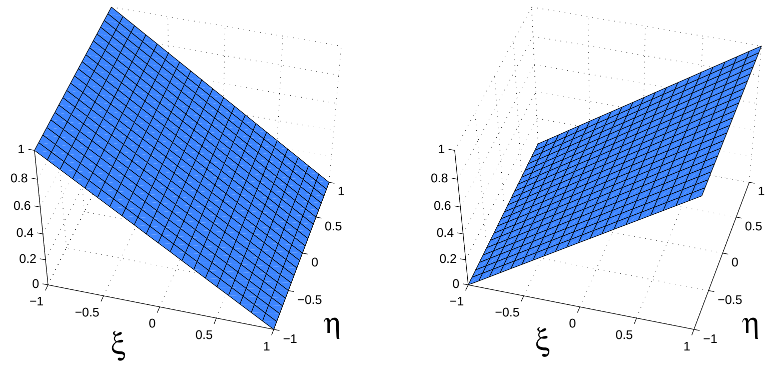

Figure 3.4. Local test functions for the second-order time integrator, in the one-dimensional case. ξ is the local space coordinate, while η is the local time coordinate.

$$\begin{aligned}\mathcal{S}^h &= \Big\{\boldsymbol{V}^h : \boldsymbol{V}^h \in (C^0(Q))^m, \\ &\qquad \boldsymbol{V}^h\big|_{Q_n^e} \in (\mathcal{P}_1(V^e) \times \mathcal{P}_1(I_n))^m, \boldsymbol{V}^h = \boldsymbol{G}_{bc}(t) \text{ on } P_n^g\Big\} \qquad (3.12)\\ \mathcal{V}^h &= \Big\{\boldsymbol{W}^h : \boldsymbol{W}^h\big|_V \in (C^0(V))^m, \\ &\qquad \boldsymbol{W}^h\big|_{Q_n^e} \in (\mathcal{P}_1(V^e) \times \mathcal{P}_0(I_n))^m, \boldsymbol{V}^h = \boldsymbol{0} \text{ on } P_n^g\Big\} \qquad (3.13)\end{aligned}$$

with $m = 2n_d + 1, \ \ n_d \in \{1, 2, 3\}$. Equation (3.8) reduces to

$$\begin{aligned}0 &= \int_V \boldsymbol{W}^h(\boldsymbol{X}) \cdot \boldsymbol{U}(\boldsymbol{Y}^h(\boldsymbol{X}, t_{n+1})) - \boldsymbol{W}^h(\boldsymbol{X}) \cdot \boldsymbol{U}(\boldsymbol{Y}^h(\boldsymbol{X}, t_n)) \ dV \\ &+ \int_{Q_n} \left(-\boldsymbol{W}^h_{,i} \cdot \boldsymbol{G}_i(\boldsymbol{Y}^h) + \boldsymbol{W}^h \cdot \boldsymbol{Z}(\boldsymbol{Y}^h)\right) \ dQ \\ &+ \int_{P_n^g} \boldsymbol{W}^h \cdot \boldsymbol{G}_i(\boldsymbol{Y}^h) N_i dP + \int_{P_n^h} \boldsymbol{W}^h \cdot \boldsymbol{H}_i N_i \ dP \\ &+ \mathcal{SUPG}(\boldsymbol{W}^h, \boldsymbol{Y}^h) + \mathcal{DC}(\boldsymbol{W}^h, \boldsymbol{Y}^h) \qquad (3.14)\end{aligned}$$

Since $\boldsymbol{Y}^h$ is *continuous* in *space* and *time*, there is no need to distinguish between positive and negative limit values for a certain instant in time. In addition, $\boldsymbol{W}^h$ is constant in time, so $\boldsymbol{W}^h = \boldsymbol{W}^h(\boldsymbol{X})$ on each space-time slab. The time derivative of $\boldsymbol{W}^h$ vanishes, leaving only the jump terms in time to represent the time-derivative operator.

Chapter 4

SUPG Stabilization

SUPG stabilization of hyperbolic systems of conservation laws is well established, and one can refer to the paper by Shakib, Hughes and Johan [44] for an extensive survey of the main techniques and corresponding references.

It is also documented in the literature that SUPG stabilization prevents *locking/artificial stiffness* for P1-elements (see, e.g., [31, 35, 36, 34]).

4.1 A multiscale perspective

There is a multiscale interpretation to the SUPG stabilization concept, which is presented next. Let us start from the variational form of the Lagrangian hydrodynamics equations (3.8), in which we assume, at the abstract level, that it is possible to have full knowledge about the exact solution $\boldsymbol{Y} \in \mathcal{S}$, with $\mathcal{S}^h \subset \mathcal{S}$. The test space counterpart of $\mathcal{S}$ is $\mathcal{V} \supset \mathcal{V}^h$. Hence, (3.8) reduces to:

$$\mathcal{B}(\boldsymbol{W}, \boldsymbol{Y}) \;=\; \mathcal{F}(\boldsymbol{W}) \tag{4.1}$$

Notice that the SUPG and DC operators have not been included, since $\mathcal{V}$ contains full information about the solution $\boldsymbol{Y}$. For the sake of simplicity, we will assume that no body forces or Neumann conditions are applied, so that $\mathcal{F}(\boldsymbol{W}) \equiv 0$, throughout.

Let us now decompose the solution into a *coarse-scale* component, or *mesh* solution $\boldsymbol{Y}^h \in \mathcal{S}^h$ (the component of $\boldsymbol{Y}$ resolved by the numerical mesh), and a *fine-scale* or *subgrid* component $\boldsymbol{Y}' \in \mathcal{S}'$, ($\mathcal{S}' = \mathcal{S} \backslash \mathcal{S}^h$ is the complement of $\mathcal{S}^h$ to $\mathcal{S}$). Analogously, the test function $\boldsymbol{W}$ can be decomposed as $\boldsymbol{W} = \boldsymbol{W}^h + \boldsymbol{W}'$, $\boldsymbol{W}' \in \mathcal{V}' = \mathcal{V} \backslash \mathcal{V}^h$. Assuming that $\mathcal{V}'$ and $\mathcal{V}^h$ are *linearly independent*, as well as $\mathcal{S}'$ and $\mathcal{S}^h$, leads to a decomposition of the original problem into two subproblems, namely:

$$\begin{aligned} \mathcal{B}(\boldsymbol{W}^h, \boldsymbol{Y}^h + \boldsymbol{Y}') &= 0 \qquad \text{(mesh-scale problem)} \qquad (4.2) \\ \mathcal{B}(\boldsymbol{W}', \boldsymbol{Y}^h + \boldsymbol{Y}') &= 0 \qquad \text{(subgrid-scale problem)} \qquad (4.3) \end{aligned}$$

The sub-grid scale equation (4.3) can be simplified by making a local linearization (for details, see [40]). On each element, the nonlinear operators are linearized about a local base solution (for example the average values of $\boldsymbol{Y}^h$ over each element). The residual operator $\boldsymbol{Res}$ is defined by means of the quasi-linear form of the Lagrangian hydrodynamics equations:

$$\boldsymbol{Res} \;=\; \mathcal{L} = \boldsymbol{A}_0 \frac{\partial}{\partial t} + \boldsymbol{A}_i \frac{\partial}{\partial X_i} + \boldsymbol{C} \qquad (4.4)$$

Equation (4.3) will then transform, after linearization, to

$$(\boldsymbol{W}', \mathcal{L}\boldsymbol{Y}') \;=\; -(\boldsymbol{W}', \mathcal{L}\boldsymbol{Y}^h) = -(\boldsymbol{W}', \boldsymbol{Res}(\boldsymbol{Y}^h)) \qquad (4.5)$$

where $(\cdot, \cdot)$ indicates the L^2 inner product over the space-time slab, and boundary terms have been omitted, for the sake of brevity. Formally, one can invert (4.5) by means of an inverse integral operator $\mathcal{L}^{-1}$, involving a Green's function kernel $\boldsymbol{G}'$

$$\boldsymbol{Y}'|_{Q_n^e}(\boldsymbol{X}) \;=\; -\int_{Q_n^e} \boldsymbol{G}' \; \boldsymbol{Res}(\boldsymbol{Y}^h) \; dQ \qquad (4.6)$$

Although this is only a formal step, it becomes apparent, after detailed derivations are carried out, that:

1. The Lagrangian hydrodynamic equations simplify, after linearization, to the wave equation. It is easy to verify the assertion in the case of compressible gas dynamics. It is well known that the linearization of the compressible Euler equations is given by the equations of acoustics, which are hyperbolic wave equations. Hence, $\boldsymbol{G}'$ can be proved to be a typical wave propagation Green's function kernel.

2. The multiscale analysis confirms the argument of Kuropatenko [32] regarding the acoustic nature of the artificial viscosity in the limit for weak shocks or, more appropriately, isentropic compressions. In this case the solution is smooth enough to be amenable to local linearization, and the presented multiscale analysis applies in a very strict sense. Detailed derivations on how the SUPG stabilization relates to the Kuropatenko analysis and the "acoustical" part of HEMP-type viscosities [4, 52] are found in [40].

Equation (4.2) is amenable to a local linearization:

$$\mathcal{B}(\boldsymbol{W}^h, \boldsymbol{Y}^h + \boldsymbol{Y}') \approx \mathcal{B}(\boldsymbol{W}^h, \boldsymbol{Y}^h) + (\mathcal{L}^* \boldsymbol{W}^h, \boldsymbol{Y}') \tag{4.7}$$

where $\mathcal{L}^*$ is the adjoint operator of $\mathcal{L}$. Equation (4.7) is used to implement the stabilization term. The definition of the SUPG operator is documented for the case of hyperbolic systems (see Shakib, Hughes and Johan [44], and the earlier paper by Hughes and Mallet [22], for a discussion), and consists in approximating the matrix Green's function $\boldsymbol{G}'$ by a tensor $\boldsymbol{\tau}$

$$\mathcal{SUPG}(\boldsymbol{W}^h, \boldsymbol{Y}^h) = -\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} (\mathcal{L}_{SH}^* \boldsymbol{W}^h) \cdot \boldsymbol{\tau}\ \boldsymbol{Res}(\boldsymbol{Y}^h) dQ \tag{4.8}$$

where $\mathcal{L}_{SH}^* = -\boldsymbol{A}_0^T \partial_t - \boldsymbol{A}_i^T \partial_i$ is the adjoint of the acoustic wave propagation differential operator $\mathcal{L}_{SH} = \boldsymbol{A}_0 \partial_t + \boldsymbol{A}_i \partial_i$ (in the SUPG literature, $\mathcal{L}_{SH}$ is termed the generalized advective operator). The definition of the tensor $\boldsymbol{\tau}$ is specific to the system of equations to be solved and a general methodology for its derivation can be found in [44]. However a strict implementation in multiple dimensions of such an approach could not be pursued successfully in the Lagrangian case, and a new design was therefore developed.

4.2 A new design requirement: Galilean invariance

It was found extremely beneficial to ensure Galilean invariance of the SUPG stabilization operator. It became apparent in early numerical simulations that lack of Galilean invariance could lead to instabilities [40].

The issue can be best understood by noticing that an SUPG method corresponds to a Petrov-Galerkin formulation in which the test function space depends on the structure of the system of partial differential equations simulated. Therefore, it is important to ensure that the test function space is invariant under Galilean transformations, otherwise the stability properties of the SUPG method may be *observer-dependent*. In particular, the perturbation to the Galerkin test space represented by $-(\mathcal{L}_{SH}^* \boldsymbol{W}^h) \cdot \boldsymbol{\tau}$ cannot be an explicit function of the material velocity $\boldsymbol{v}$, unless $\boldsymbol{v}$ appears in differentiated form.

In addition, it easy to realize that the subgrid-scale solution must be Galilean invariant, since it is the difference of the vectors $\boldsymbol{Y}$ and $\boldsymbol{Y}^h$. Therefore, it is also reasonable to impose Galilean invariance for the approximation $\boldsymbol{Y}' \approx \boldsymbol{\tau}\ \boldsymbol{Res}(\boldsymbol{Y}^h)$. If both the perturbation to the test function and the approximation to $\boldsymbol{Y}'$ are to be invariant, then the overall SUPG operator results invariant.

As discussed in detail in [40], invariance can be achieved by removing the kinetic energy equation from the total energy equation before deriving the quasi-linear form.

4.3 Stabilization in the Lagrangian framework

In the framework of Lagrangian hydrodynamics it is possible to simplify the expressions for the SUPG operator, since stabilization affects only the momentum and energy equations. This can be easily seen by carrying over the calculation of stabilization terms by brute force in the one-dimensional case, for which the approach in [44] applies successfully. The rationale is that the kinematic equations relating the rates of displacement to the velocities are actually ODEs in the degrees-of-freedom of the discrete solution, so that stabilization - peculiar to boundary value problems for PDEs - is not needed. It is possible then to reformulate the structure of the matrix $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{0}_{n_d \times n_d} & \mathbf{0}_{n_d \times (n_d+1)} \\ \mathbf{0}_{(n_d+1) \times n_d} & \hat{\boldsymbol{\tau}}_{(n_d+1) \times (n_d+1)} \end{bmatrix} \tag{4.9}$$

As it is easily realized, in the case of one, two and three dimensions in space, instead of developing a $\boldsymbol{\tau}$ tensor of size 3×3, 5×5, or 7×7, it is sufficient to compute just the $\hat{\boldsymbol{\tau}}$ tensor of size 2×2, 3×3, or 4×4, respectively.

Remark 6 *Since stabilization is applied only to the momentum and energy equations, it is easy to verify that* $\hat{\mathcal{L}}^* = \hat{\mathcal{L}}^*_{SH}$.

The newly developed $\hat{\boldsymbol{\tau}}$ tensor reads:

$$\hat{\boldsymbol{\tau}} = \frac{\Delta t}{2\, CFL} \boldsymbol{A}_0^{-1} = \frac{\min_{1 \leq j \leq n_{el}} (\Delta x_j / (c_s)_j)}{2} \boldsymbol{A}_0^{-1} \tag{4.10}$$

Remark 7 $CFL = \Delta t \max_{1 \leq j \leq n_{el}} ((c_s)_j / \Delta x_j)$ *is the* global *Courant-Friedrichs-Levy number,* $(c_s)_j$ *is the speed of sound,* Δt *is the time increment and* Δx_j *is a characteristic mesh scale in the current configuration, such as the smallest distance between nodes belonging to the same element. The term* $\Delta t/(2CFL) = \min_{n_{el}} (\Delta x/(2c_s))$ *does not vanish as* Δt *tends to zero, preventing the stabilization term from decaying too rapidly for small time steps. Such issues have been extensively discussed, for example, in [5] and [11].*

Remark 8 $\boldsymbol{A}_0$ *is diagonal, allowing for a fast computation of the stabilization term, compared to [44], in which the inverse square root of a tensor has to be evaluated or approximated, in the case of multiple dimensions.*

Remark 9 *The stabilization matrix* $\boldsymbol{\tau}$ *has been devised as a very simple generalization of the one-dimensional matrix obtained by the approach described in [44]. The details of the one-dimensional computations are in [41], and lead to the expression*

$$\hat{\boldsymbol{\tau}} = \frac{\Delta t}{2\sqrt{1+\alpha^2}} \boldsymbol{A}_0^{-1} \tag{4.11}$$

where $\alpha = \Delta t \ (c_s)_j / \Delta x_j$ *is the* local *CFL number.*

Chapter 5

Discontinuity capturing operator

The design of the Discontinuity Capturing (DC) operator is essential to the overall performance of the numerical method. Shocks must be smoothed in a band of possibly no more than 3-4 elements. Three key elements are the basis for the design of the proposed DC operator:

1. A von Neumann-Richtmyer [51] artificial viscosity was adopted with an additional artificial heat flux analogous to the one proposed by Noh [39].
2. The computation of the mesh scale parameter present in the von Neumann-Richtmyer formula is performed by combining the approach of Wilkins [52] and Tezduyar [48, 47, 49]. Special care was taken to avoid abrupt element-to-element variations of the mesh scale parameter because of negative consequences on the artificial viscosity performance. This concept is presented in detail in [9].
3. It was also found very important to have the viscosity peak on the elements leading the shock layer, as the shock moves through the material. This allowed for improved smoothness of the solution across the shock layer and a reduction of the viscosity coefficient.

In spite of not being particularly sophisticated compared to the latest concepts in the field (see, e.g., [4, 9, 8] for advanced concepts on TVD limiting), the proposed viscosity proves robust and effective even in the more demanding tests. Most of the advanced concepts to date use edge-centered viscosities, easily implemented in the context of finite-difference or finite-volume methods, but less straightforward to implement for a standard finite element method. It was felt to be beyond the scope of the present paper to investigate such concepts. These may be the focus of future research.

Remark 10 *Although improvements can be made on the viscosity, the numerical results are in most cases equal or sometimes superior to state-of-the-art techniques for* *both triangular and quadrilateral meshes. Apparently, improved gradient representation and SUPG stabilization seem to have a very significant effect on the overall quality of the results.*

The discontinuity capturing operator is implemented as follows:

$$\begin{aligned}
\mathcal{D}C(\boldsymbol{W}^h, \boldsymbol{Y}^h) \quad = \quad & \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \mathbf{Grad}\boldsymbol{W}^h_{\boldsymbol{v}} : \boldsymbol{P}_{art}\; dQ && (5.1) \\
& + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \mathbf{Grad} W^h_E \cdot \left(\boldsymbol{P}^T_{art}\boldsymbol{v} + \boldsymbol{Q}_{art} \right)\; dQ \\
& - \int_{P_n} W^h_E \left(\boldsymbol{P}^T_{art}\boldsymbol{v} \cdot \boldsymbol{N} \right)\; dP && (5.2)
\end{aligned}$$

where

$$\boldsymbol{P}_{art} \quad = \quad J\boldsymbol{\sigma}_{art}\boldsymbol{F}^{-T} = \boldsymbol{\sigma}_{art}(\mathbf{cof}\boldsymbol{F}) \qquad (5.3)$$

$$\boldsymbol{Q}_{art} \quad = \quad J\boldsymbol{F}^{-1}\boldsymbol{q}_{art} = (\mathbf{cof}\boldsymbol{F})^T\boldsymbol{q}_{art} \qquad (5.4)$$

and

$$\boldsymbol{\sigma}_{art} \quad = \quad \rho\; \nu_{art}\; \frac{1}{2}\left(\mathbf{grad}\; \boldsymbol{v} + \mathbf{grad}^T \boldsymbol{v} \right) \qquad (5.5)$$

$$\boldsymbol{q}_{art} \quad = \quad \rho\; \nu_{art}\; g_2\left(\rho_0/J, p\right)\; \mathbf{grad}\; p \qquad (5.6)$$

$$\nu_{art} \quad = \quad C_\nu\; f_*(\dot{J})\; h^2_{\boldsymbol{b}}\; |\mathbf{div}\; \boldsymbol{v}|\; \chi(\{\mathbf{div}\; \boldsymbol{v} < 0\}) \qquad (5.7)$$

where $\chi(\{\mathbf{div}\; \boldsymbol{v} < 0\})$ is the characteristic function of the set $\{\mathbf{div}\; \boldsymbol{v} < 0\}$ and $C_\nu = 2.0$. $\boldsymbol{W}^h_{\boldsymbol{v}}$ is the test function vector relative to the momentum equations only, and W^h_E is the scalar test function relative to the energy equation. $\boldsymbol{P}_{art}$ and $\boldsymbol{Q}_{art}$ are the Piola transformation of the artificial stress tensor $\boldsymbol{\sigma}_{art}$ and artificial heat flux vector $\boldsymbol{q}_{art}$. $\boldsymbol{\sigma}_{art}$ is a function of the symmetric part of the velocity gradient in the current configuration, which ensures invariance under rotations and Galilean transformations. $\boldsymbol{q}_{art}$ is a function of the gradient of the internal energy, since the pressure gradient is scaled with the term $\partial e/\partial p = \rho_0 g_2\left(\rho_0/J, p\right)$. f_* is a term depending on the element type:

$$f_* = \begin{cases} 1, & \text{for segments, quadrilaterals, hexahedra} \\ 1 + e^{-\beta \frac{\partial}{\partial t}(\ln J)}, & \text{for triangles, tetrahedra} \end{cases} \qquad (5.8)$$

In practice, the derivative of the natural logarithm of J further shifts the peak of artificial viscosity ahead of the shock for triangles and tetrahedra, increasing the smoothness of the shock profile. $\beta = 0.75$ was found an appropriate choice. The time derivative is easily computed by means of the predictor-corrector time integration strategy developed in chapter 7.

Remark 11 *Since f_* is a function of $\dot{J}$, it maintains the invariance and objectivity properties of the artificial viscosity operator.*

Remark 12 *The term $\boldsymbol{P}_{art}^T \boldsymbol{v}$, or artificial stress dissipation is usually not incorporated in state-of-the-art viscosities for aerospace computations (see [17, 44, 48, 47, 49]), but it turns out to be extremely important in implosion computations when shocks are generated by moving boundaries (e.g., piston-type boundary conditions).*

Remark 13 *Notice also that a boundary integral for the dissipation $\boldsymbol{P}_{art}^T \boldsymbol{v}$ has been added. Using integration by parts, it is easily understood that such term contributes to the discretization of the divergence of $\boldsymbol{P}_{art}^T \boldsymbol{v}$. A similar approach would apply to $\boldsymbol{P}_{art}$ in the momentum equation and to $\boldsymbol{Q}_{art}$ in the energy equation, but such additional contributions were found to negatively affect numerical results, and were therefore omitted.*

Remark 14 *A considerable literature is available on discontinuity capturing operators for finite element methods. Such operators usually take the form of a purely residual-based artificial viscosity, rapidly vanishing when the solution is smooth, as for isentropic compressions/expansions. This is a very useful property, but attempts made to design a residual-based viscosity for Lagrangian hydrodynamics simulations have not been successful to date. The main problem related to the residual-based viscosities tested was their rapid variation from element to element, which undermines their potential in the case of transient shock propagation computations. Further investigations will be directed toward this issue.*

Remark 15 *Special care needs to be devoted to the definition of $h_{\boldsymbol{b}}$, the mesh scaling, in the definition of the artificial viscosity, as documented in the next section.*

5.1 The mesh parameter "$h_{\boldsymbol{b}}$"

The definition of the mesh parameter can be expressed as

$$\tilde{h}_{\boldsymbol{b}} = 0.75 \left(2\|\boldsymbol{b}\|_{l_2} \left(\sum_{a=1}^{n_{en}} |\boldsymbol{b} \cdot \mathbf{grad}\ N_a(\boldsymbol{x})| \right)^{-1} \right) \tag{5.9}$$

$$+0.25 \left(2\|\dot{\boldsymbol{v}}\|_{l_2} \left(\sum_{a=1}^{n_{en}} |\dot{\boldsymbol{v}} \cdot \mathbf{grad}\ N_a(\boldsymbol{x})| \right)^{-1} \right) \tag{5.10}$$

$$\boldsymbol{b} = \mathbf{grad}\|\boldsymbol{v} - \boldsymbol{v}_{\min}^{(e)}\|_{l^2} \tag{5.11}$$

This definition is a compromise between earlier definitions by Wilkins [52], in the context of hydrocode simulations for structural mechanics, and Tezduyar, in the context of SUPG methods for the compressible Navier-Stokes equations (see, e.g., [48, 47, 49]). Effectively, (5.10) is a linear combination of the length scale in the direction of the local gradient of the magnitude of the velocity and the local acceleration direction.

Remark 16 *The vector* $\boldsymbol{v}_{\min}^{(e)}$ *is the velocity vector of minimum norm over the element e, namely*

$$\boldsymbol{v}_{\min}^{(e)} = \mathrm{argmin}_{1 \le n \le n_{np}^{e}} \|\boldsymbol{v}_n^{(e)}\|_{l^2} \tag{5.12}$$

where n_{np}^{e} *is the number of nodes of the element e. Subtracting* $\boldsymbol{v}_{\min}^{(e)}$ *from* $\boldsymbol{v}$ *in (5.11) is consistent with Galilean invariance and improves the accuracy in detecting the direction normal to the shock front for elements of poor aspect ratio.*

Remark 17 *Both vectors* $\boldsymbol{b}$ *and* $\dot{\boldsymbol{v}}$ *are used to define* $\tilde{h}_{\boldsymbol{b}}$*, since the acceleration direction was found superior in detecting the normal to the shock front, while the vector* $\boldsymbol{b}$ *was found more reliable in stabilizing mild but persistent oscillations near moving boundaries, a common feature of hydrocode simulations.*

5.1.1 A limiting strategy for the mesh parameter

One of the important aspects in the design of artificial viscosities, is to make sure they vary smoothly across shock fronts, otherwise some of the elements may become over-damped compared with their closest neighbors, with very negative effects on the quality of the results. A simple *limiter* acting on $\tilde{h}_{\boldsymbol{b}}$ was designed. A more general approach would be to apply a limiting technique directly to the viscosity (see, e.g.,

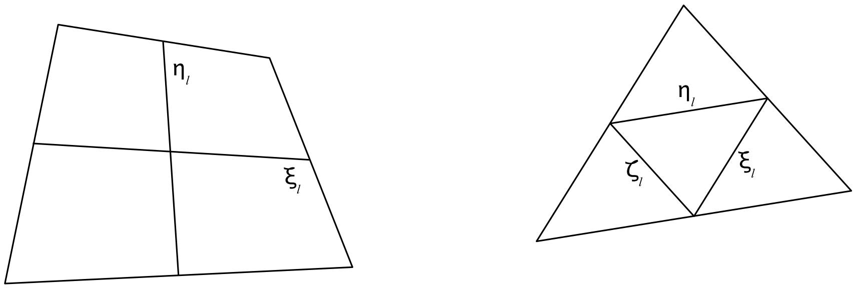

Figure 5.1. Sketch of the segments connecting the mid-points for quadrilaterals (left) and triangles (right).

[4, 9, 8]), which might improve results. However, one of the appealing features of the currently implemented limiting on $\tilde{h}_{\boldsymbol{b}}$ is its locality in terms of the element data structure. For this purpose, it will be useful to define the vector:

$$\boldsymbol{u_b} = \frac{0.75\boldsymbol{b} + 0.25\dot{\boldsymbol{v}}}{\|0.75\boldsymbol{b} + 0.25\dot{\boldsymbol{v}}\|_{l^2}} \tag{5.13}$$

5.1.2 Quadrilaterals

In the case of quadrilaterals, the limiting consists in taking the segments connecting the mid-point of opposite edges of the quadrilateral ($\boldsymbol{\xi}_l$ and $\boldsymbol{\eta}_l$ on the left sketch of Fig. 5.1), and enforcing the following constraint:

$$h_{\boldsymbol{b}} = \min\left(\tilde{h}_{\boldsymbol{b}}, \sqrt{\left(\frac{\cos(\theta_l)}{\|\boldsymbol{\xi}_l\|_{l^2}}\right)^2 + \left(\frac{\sin(\theta_l)}{\|\boldsymbol{\eta}_l\|_{l^2}}\right)^2}\right) \tag{5.14}$$

$$\theta_l = \frac{\pi}{2}\frac{\arccos(|\boldsymbol{u_b} \cdot \boldsymbol{\xi}|)/\|\boldsymbol{\xi}_l\|_{l^2})}{\arccos(|\boldsymbol{\xi} \cdot \boldsymbol{\eta}|)/(\|\boldsymbol{\xi}_l\|_{l^2}\|\boldsymbol{\eta}_l\|_{l^2})} \tag{5.15}$$

In practice, $h_{\boldsymbol{b}}$ is bounded by the contour of an ellipse of semi-axes $\boldsymbol{\xi}_l/2$ and $\boldsymbol{\eta}_l/2$.

5.1.3 Triangles

In the case of triangles, the segments $\boldsymbol{\xi}_l$, $\boldsymbol{\eta}_l$, and $\boldsymbol{\zeta}_l$ connecting the mid-points of the triangular element edges (see Fig. 5.1) are used to limit $h_{\boldsymbol{b}}$ according to

$$h_{\boldsymbol{b}} = \min\left(\tilde{h}_{\boldsymbol{b}}, \operatorname{argmin}_{\boldsymbol{w} \in \{\boldsymbol{\xi}_l, \boldsymbol{\eta}_l, \boldsymbol{\zeta}_l\}}\left(|\tilde{h}_{\boldsymbol{b}} \cdot \boldsymbol{w}|/\|\boldsymbol{w}\|_{l^2}\right)\right) \tag{5.16}$$

In this case, $h_{\boldsymbol{b}}$ is bounded by its maximal projection on $\boldsymbol{\xi}_l$, $\boldsymbol{\eta}_l$, or $\boldsymbol{\zeta}_l$ (in the l^2-inner product sense).

Chapter 6

Global conservation

The proposed formulation is globally conservative. We assume $P_n^g = \emptyset$, as is commonly done in the derivation of conservation laws (see Hughes *et al.* [19]). Let us set to unity over a time-slab one of the entries of the test vector $\boldsymbol{W}^h$ corresponding to the momentum or energy equations, while keeping zero all remaining entries. Namely, $\boldsymbol{W}_i^h = 1$, for some $i \in \{n_d+1, \ldots, 2n_d+1\}$, and $\boldsymbol{W}_j^h = 0$ if $j \neq i$. This choice is made possible by the fact that $\boldsymbol{W}^h$ is piecewise continuous linear in space and constant discontinuous in time, so it can represent global constants. Equation (3.8) reduces to:

$$\begin{aligned} 0 \;=\; & \int_V \boldsymbol{U}_i(\boldsymbol{Y}^h(\boldsymbol{X}, t_{n+1}^-))\, dV - \boldsymbol{U}_i(\boldsymbol{Y}^h(\boldsymbol{X}, t_n^-))\, dV \\ & + \int_{P_n^h} \boldsymbol{H}_i N_i \, dP \end{aligned} \tag{6.1}$$

which is a statement of conservation from time t_n^- to time t_{n+1}^- for the $\boldsymbol{U}_i$ entry of the conservation variables vector $\boldsymbol{U}$, provided the boundary integral vanishes. Indeed, (6.1) is readily obtained, since $\boldsymbol{W}_{,t}^h$ and $\boldsymbol{W}_{,i}^h$ vanish and so do the $\mathcal{SUPG}(\boldsymbol{W}^h, \boldsymbol{Y}^h)$ and $\mathcal{DC}(\boldsymbol{W}^h, \boldsymbol{Y}^h)$ operators, by definition. Finally, $\boldsymbol{Z}(\boldsymbol{Y}^h)$, has zero entries for the momentum and energy equation, since it is not a true source term, but, rather, a kinematic term containing the components of the velocity field for the displacement equations.

Chapter 7

Explicit predictor/multi-corrector algorithm

Lagrangian hydrodynamics algorithms typically adopt explicit time-marching schemes, since ideally, for time-accuracy purposes, the algorithms should run time-steps very close to $CFL = 1$, in accordance with a linearized analysis. When explicit algorithms are used for complex nonlinear systems, often nonlinear effects can further limit the CFL values. On the other end, implicit time-marching schemes become effective when run at significantly larger CFL values, and although examples of implicit hydrocodes are present (e.g., in contact dynamics applications), usually explicit approaches are preferred.

An explicit, predictor/multi-corrector scheme can be easily defined for the proposed method, by modifying a standard implicit Newton nonlinear iterative solver. In fact, if only the local integrals contributing to a lumped mass matrix are assembled into a *modified* Newton tangent matrix, no linear system solves are required as the iteration proceeds. A detailed explanation of the theoretical framework for this class of time-integration algorithms is presented in Hughes [18], p. 562, while the paper by Shakib, Hughes and Johan [44] is very useful for an example of the implementation in the context of compressible flow computations for aerospace applications in Eulerian coordinates.

7.1 The general nonlinear system

On a single time-slab, the solution and test vector functions can be expressed as:

$$
\begin{aligned}
\boldsymbol{Y}^h(\boldsymbol{X},t) &= \sum_{A=1}^{n_{np}} \boldsymbol{N}_A(\boldsymbol{X}) \left(\pi_n^+(t)\boldsymbol{y}_{A;(n+1)} + \pi_n^-(t)\boldsymbol{y}_{A;(n)}\right) && (7.1)\\
\boldsymbol{W}^h(\boldsymbol{X},t) &= \sum_{A=1}^{n_{np}} \boldsymbol{N}_A(\boldsymbol{X})\boldsymbol{w}_{A;(n+1)} && (7.2)
\end{aligned}
$$

with $\boldsymbol{X} \in V, t \in I_n$, $\pi_n^+(t) = (t - t_n)/\Delta t$, and $\pi_n^-(t) = (t_{n+1} - t)/\Delta t$. Also, let

$$
\begin{aligned}
(\mathcal{L}^*_{adv})^h_A &= -\boldsymbol{A}_i^T \boldsymbol{N}_{A,i} && (7.3)\\
(\mathcal{L}^h_\pm)_A &= \boldsymbol{A}_0 \boldsymbol{N}_A \pi^\pm_{,t} + \boldsymbol{A}_i \boldsymbol{N}_{A,i}\pi^\pm + \boldsymbol{C}\boldsymbol{N}_A\pi^\pm && (7.4)
\end{aligned}
$$

Substitution of (7.1)–(7.4) into (3.14) yields

$$
\begin{aligned}
0 = \sum_{A=1}^{n_{np}} \boldsymbol{w}_A \cdot \Bigg\{ & \int_V \boldsymbol{N}_A\, \boldsymbol{U}\!\left(\sum_{B=1}^{n_{np}} \boldsymbol{N}_B \boldsymbol{y}_{B;(n+1)}\right) - \boldsymbol{N}_A \boldsymbol{U}\!\left(\sum_{B=1}^{n_{np}} \boldsymbol{N}_B \boldsymbol{y}_{B;(n)}\right)\, dV \\
& - \int_{Q_n} \boldsymbol{N}_{A,i} \boldsymbol{G}_i\!\left(\sum_{B=1}^{n_{np}} \boldsymbol{N}_B \left(\pi_n^+ \boldsymbol{y}_{B;(n+1)} + \pi_n^- \boldsymbol{y}_{B;(n)}\right)\right)\, dQ \\
& + \int_{Q_n} \boldsymbol{N}_A \boldsymbol{C} \sum_{B=1}^{n_{np}} \boldsymbol{N}_B \left(\pi_n^+ \boldsymbol{y}_{B;(n+1)} + \pi_n^- \boldsymbol{y}_{B;(n)}\right)\, dQ \\
& + \int_{P_n^g} \boldsymbol{N}_A \cdot \boldsymbol{G}_i\!\left(\sum_{B=1}^{n_{np}} \boldsymbol{N}_B \left(\pi_n^+ \boldsymbol{y}_{B;(n+1)} + \pi_n^- \boldsymbol{y}_{B;(n)}\right)\right) N_i\, dP \\
& + \int_{P_n^h} \boldsymbol{N}_A \cdot \boldsymbol{H}_i N_i\, dP \\
& - \sum_{e=1}^{n_{el}} \int_{Q_n^e} (\mathcal{L}^*_{adv})^h_A \cdot \boldsymbol{\tau} \sum_{B=1}^{n_{np}} \left((\mathcal{L}^h_+)_B \boldsymbol{y}_{B;(n+1)} + (\mathcal{L}^h_-)_B \boldsymbol{y}_{B;(n)}\right)\, dQ^e_n \\
& + \mathcal{DC}\left(\boldsymbol{N}_A, \sum_{B=1}^{n_{np}} \boldsymbol{N}_B \left(\pi_n^+ \boldsymbol{y}_{B;(n+1)} + \pi_n^- \boldsymbol{y}_{B;(n)}\right)\right)\Bigg\} \qquad (7.5)
\end{aligned}
$$

7.2 Newton solver

Since on the interval I_n the nodal values $\boldsymbol{y}_{B;(n)}$ are known from the previous time-step, the system can be abstractly represented as

$$
\boldsymbol{w} \cdot \widetilde{\boldsymbol{NL}}(\boldsymbol{y}_{(n+1)}; \boldsymbol{y}_{(n)}) = 0 \qquad (7.6)
$$

and, since (7.6) has to hold for any $\boldsymbol{w}$, we obtain

$$\widetilde{\boldsymbol{NL}}(\boldsymbol{y}_{(n+1)}; \boldsymbol{y}_{(n)}) = \boldsymbol{0} \tag{7.7}$$

The Newton iteration reads:

$$\widetilde{\boldsymbol{NL}}(\boldsymbol{y}_{(n+1)}; \boldsymbol{y}_{(n)}) \approx \widetilde{\boldsymbol{NL}}(\boldsymbol{y}^{(i)}; \boldsymbol{y}_{(n)}) + \frac{\partial \widetilde{\boldsymbol{NL}}(\boldsymbol{y}^{(i)}; \boldsymbol{y}_{(n)})}{\partial \boldsymbol{y}} \, \delta \boldsymbol{y} = \boldsymbol{0} \tag{7.8}$$

with $\delta \boldsymbol{y} = (\boldsymbol{y}^{(i+1)} - \boldsymbol{y}^{(i)})$. Here the upper index (i) refers to an iterate of the Newton algorithm, while $\boldsymbol{y}_{(n+1)}$ and $\boldsymbol{y}_{(n)}$ refer to the solutions at times t_{n+1} and t_n, respectively. Starting from (7.8) we can also accommodate boundary conditions of Dirichlet type, by removing the rows of $\widetilde{\boldsymbol{NL}}$ corresponding to nodes on the Dirichlet part of the boundary. Equation (7.8) can be therefore rewritten as:

$$\boldsymbol{M}^{*^{(i)}} \, \delta \boldsymbol{y} = -\boldsymbol{R}^{(i)} \tag{7.9}$$

where the more convenient notation

$$\begin{aligned} \boldsymbol{M}^{*^{(i)}} &= \partial_{\boldsymbol{y}} \boldsymbol{NL}(\boldsymbol{y}^{(i)}; \boldsymbol{y}_{(n)}) && (7.10) \\ \boldsymbol{R}^{(i)} &= \boldsymbol{NL}(\boldsymbol{y}^{(i)}; \boldsymbol{y}_{(n)}) && (7.11) \end{aligned}$$

has been used, and $\boldsymbol{NL}$ and $\partial_{\boldsymbol{y}} \boldsymbol{NL}$ are obtained from $\widetilde{\boldsymbol{NL}}$ and $\partial_{\boldsymbol{y}} \widetilde{\boldsymbol{NL}}$, respectively, after Dirichlet boundary conditions have been accounted for, and $\boldsymbol{R}$ is the *algebraic* residual vector.

7.3 Assembly

$\boldsymbol{M}^*$ and $\boldsymbol{R}$ are assembled in the usual way, namely:

$$\begin{aligned} \boldsymbol{R}^{(i)} &= \mathop{\mathsf{A}}_{e=1}^{n_{el}} \boldsymbol{R}^{e;(i)} && (7.12) \\ \boldsymbol{M}^{*^{(i)}} &= \mathop{\mathsf{A}}_{e=1}^{n_{el}} \boldsymbol{M}^{e;(i)} && (7.13) \end{aligned}$$

where A is the finite element assembly operator (see, e.g., [2, 18]), and $\boldsymbol{R}^{e;(i)}$, $\boldsymbol{M}^{e;(i)}$ are the element contributions to the residual and tangent matrix, respectively. The solution can be expressed, in space-time, as:

$$\boldsymbol{Y}^{e;(i)}(\boldsymbol{X}, t) = \sum_{a=1}^{n_{en}} \boldsymbol{N}_a^e(\boldsymbol{X}) \left(\pi_n^{e;+}(t) \boldsymbol{y}_a^{(i)} + \pi_n^{e;-}(t) \boldsymbol{y}_{a;(n)} \right) \tag{7.14}$$

where $\boldsymbol{N}_a^e$, is the local test function in space (constant in time on the space-time slab under consideration), and $\pi_n^{e;\pm}$ are the local trial functions in time. With the previous assumptions,

$$\begin{aligned}
\boldsymbol{R}^{e;(i)} &= \{\boldsymbol{R}_a^{e;(i)}\} \qquad (7.15)\\
\boldsymbol{R}_a^{e;(i)} &= \int_{V^e} \boldsymbol{N}_a^e\, \boldsymbol{U}\Big(\boldsymbol{Y}^{e;(i)}(\boldsymbol{X},t_{n+1})\Big) - \boldsymbol{N}_a^e\, \boldsymbol{U}\Big(\boldsymbol{Y}^{e;(i)}(\boldsymbol{X},t_n)\Big)\ dV\\
&+ \int_{Q_n^e} -\boldsymbol{N}_{a,i}^e \boldsymbol{G}_i\Big(\boldsymbol{Y}^{e;(i)}(\boldsymbol{X},t)\Big) + \boldsymbol{N}_a^e \boldsymbol{C}\ \boldsymbol{Y}^{e;(i)}(\boldsymbol{X},t)\ dQ\\
&+ \int_{P_n^{(g;e)}} \boldsymbol{N}_a^e \cdot \boldsymbol{G}_i\Big(\boldsymbol{Y}^{e;(i)}(\boldsymbol{X},t)\Big)\, N_i\ dP + \int_{P_n^{(h;e)}} \boldsymbol{N}_a^e \cdot \boldsymbol{H}_i N_i\ dP\\
&+ \int_{Q_n^e} \boldsymbol{A}_j^T \boldsymbol{N}_{a,j}^e \cdot \boldsymbol{\tau}\left(\boldsymbol{A}_0 \boldsymbol{Y}_{,t}^{e;(i)} + \boldsymbol{A}_k \boldsymbol{Y}_{,k}^{e;(i)} + \boldsymbol{C}\boldsymbol{Y}^{e;(i)}\right)\ dQ\\
&+ \mathcal{DC}^e(\boldsymbol{N}_a^e, \boldsymbol{Y}^{e;(i)}) \qquad (7.16)
\end{aligned}$$

where a denotes a local node number for element e. In the explicit variant considered here, a simple approximation of $\boldsymbol{M}^{*^{(i)}}$ is used. In this case,

$$\boldsymbol{M}^{*^{(i)}} \approx \boldsymbol{M}^{L^{(i)}} = \mathop{\mathsf{A}}_{e=1}^{n_{el}} \left[\delta_{ab} \int_{V^e} \boldsymbol{N}_a^e \boldsymbol{A}_0\ dV\right] \qquad (7.17)$$

where δ_{ab} is the Kronecker delta. The algorithm is summarized in Table 7.1. It is important to observe that if single-point quadrature is used for the integration in time, a mid-point integration scheme is recovered. This version of the more general space-time algorithm has been used in all the computations, where, typically, three iterations of the predictor/multi-corrector were used.

Segregating the unknowns, the general form of the iteration is

$$\begin{bmatrix} \boldsymbol{M}_{\boldsymbol{uu}}^L & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{M}_{\boldsymbol{vv}}^L & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{M}_{pp}^L \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{u}^{(i)} \\ \delta\boldsymbol{v}^{(i)} \\ \delta\boldsymbol{p}^{(i)} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{u}}^{(i)} \\ \boldsymbol{R}_{\boldsymbol{v}}^{(i)} \\ \boldsymbol{R}_{\boldsymbol{p}}^{(i)} \end{bmatrix} \qquad (7.18)$$

where $\boldsymbol{M}_{\boldsymbol{uu}}^L$, $\boldsymbol{M}_{\boldsymbol{vv}}^L$ and $\boldsymbol{M}_{pp}^L$ are diagonal.

7.3.1 Time-integration strategy for displacements

For the displacement equations,

$$\delta\boldsymbol{u}^{(i)} = -(\boldsymbol{M}_s^L)^{-1}\boldsymbol{M}_s\left(\boldsymbol{u}^{(i)} - \boldsymbol{u}_n - \frac{\Delta t_n}{2}\left(\boldsymbol{v}^{(i)} + \boldsymbol{v}_n\right)\right) \qquad (7.19)$$

```
Retrieve loop parameters: $n_{\text{step}}$, $i_{\max}$
Initialize: set $\boldsymbol{y}_{(0)}$
For $n = 0, \ldots, n_{\text{step}}$ (Time-step loop begins)
    Predictor: $\boldsymbol{y}^{(0)} = \boldsymbol{y}_{(n)}$
    Set $\Delta t$ (respecting the CFL condition)
    For $i = 0, \ldots, i_{\max} - 1$ (Multi-corrector loop begins)
        Form $\boldsymbol{R}^{(i)}(\boldsymbol{y}^{(i)}; \boldsymbol{y}_{(n)})$
        Form $\boldsymbol{M}^{L^{(i)}}(y^{(i)})$
        Update $\delta\boldsymbol{y}^{(i)}$: $\boldsymbol{M}^{L^{(i)}}\delta\boldsymbol{y}^{(i)} = -\boldsymbol{R}^{(i)}(\boldsymbol{y}^{(i)}; \boldsymbol{y}_{(n)})$
        Corrector: $\boldsymbol{y}^{(i+1)} = \boldsymbol{y}^{(i)} + \delta\boldsymbol{y}^{(i)}$
    End (Multi-corrector loop ends)
    Time update: $\boldsymbol{y}_{(n+1)} = \boldsymbol{y}^{i_{\max}}$
End (Time-step loop ends)
Exit
```

Table 7.1. Outline of the predictor-multicorrector algorithm. Three iterations of the predictor/multi-corrector were used in the computations.

where $\boldsymbol{M}_s$ indicates the consistent mass matrix for the displacements, and $\boldsymbol{M}_s^L$ is its lumped version.

Remark 18 *The much simpler approach of solving a set of ODEs for the nodal displacements, namely,*

$$\delta\boldsymbol{u}^{(i)} \quad = \quad -\left(\boldsymbol{u}^{(i)} - \boldsymbol{u}_n - \frac{\Delta t_n}{2}\left(\boldsymbol{v}^{(i)} + \boldsymbol{v}_n\right)\right) \tag{7.20}$$

produced very poor results for the density profiles in the most demanding simulations. Early attempts proved this approach too inaccurate to be further pursued.

Chapter 8

General considerations on implementation

A number of additional issues need to be addressed before proceeding with the analysis of the numerical results.

8.1 Post-processed variables

The algorithm detailed in Table 7.1 directly computes the displacements, velocities and pressures. If other quantities of interest are to be computed, the complete solution update strategy is summarized as follows:

1. The primary variables $\boldsymbol{Y} = [\boldsymbol{u},\ \boldsymbol{v},\ p]^T$ are solved for numerically, using the procedure detailed in Table 7.1.

2. The determinant $J = \det \boldsymbol{F} = [\partial x_i / \partial X_j]$ of the deformation gradient is computed from the displacement $\boldsymbol{u}$.

3. The current configuration density $\rho = \rho_0 / J$ is determined.

4. The internal energy e is determined using the equation of state (2.11).

Note that energy and density are not directly computed, but are determined by "post-processing" the displacements and pressures. Any time they appear in the variational form, their expressions in terms of the initial density ρ_0, the Jacobian J, the pressure p and the specific heat ratio γ are used.

8.2 Proper specification of the density initial condition

In order to start the computations, the initial conditions have to be set. A cell-centered, piecewise constant approximation for the initial density ρ_0 proved the most accurate option, while displacements, velocities and pressure are piecewise-(multi-)linear continuous functions, with degrees-of-freedom centered at the nodes.

Because a finite element approximation is adopted, the matching of pressures and densities in the initial conditions is best accomplished in an L^2 (weak) sense, rather than point-wise.

For this purpose, a single-point quadrature L^2-projection technique was used. This technique is sketched in Figure 8.1:

1. pressure and density are initially considered constant over each element. Namely:

$$p_0^C(\boldsymbol{X}) = \sum_{e=1}^{n_{el}} p_{0;e}\chi(V_e) \tag{8.1}$$

$$\rho_0^C(\boldsymbol{X}) = \sum_{e=1}^{n_{el}} \rho_{0;e}\chi(V_e) \tag{8.2}$$

   where $\chi(V_e)$ is the characteristic function relative to the element domain V_e.

2. Pressure and density are projected using the lumped mass matrix $\boldsymbol{M}^L_{\boldsymbol{uu}}$ onto the space of continuous nodal functions, and a node-centered approximation to *both* is generated. More precisely:

$$p_{0;A}^N = \frac{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A\, p_0^C(\boldsymbol{X})\, dV\right)}{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A\, dV\right)} = \frac{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A\, p_{0;e}\, dV\right)}{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A\, dV\right)} \tag{8.3}$$

$$\rho_{0;A}^N = \frac{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A\, \rho_0^C(\boldsymbol{X})\, dV\right)}{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A\, dV\right)} = \frac{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A\, \rho_{0;e}\, dV\right)}{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A\, dV\right)} \tag{8.4}$$

   where $p_{0;A}^N$ and $\rho_{0;A}^N$ are the initial pressure and density at node A (in the global node numbering), respectively. The integrals in (8.3–8.4) are computed using a single-point quadrature.

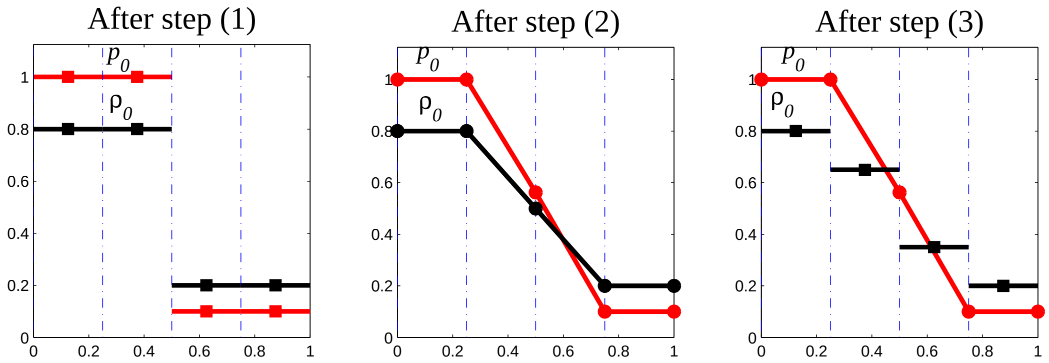

Figure 8.1. Sketch of the preprocessing procedure for pressures and densities in one space dimension.

3. The nodal density is projected back onto the elements, taking its area-weighted average over each element. Therefore, (8.2) now holds with

$$\rho_{0;e} \quad = \quad \frac{1}{V_e}\int_{V_e}\sum_{a=1}^{n_{np}^e}\rho_{0;a}^N N_a \; dV \tag{8.5}$$

   where the the subscript a refers to the local, element node numbering, contrasted with the global node index A in (8.4). Again, the integrals in (8.5) are computed using a single-point quadrature.

Remark 19 *The described pre-processing strategy for the density conserves the total mass.*

8.3 Post-processing of the current configuration density

The issue of pressure/density matching manifests itself again when an accurate approximation for the current configuration density is sought. In this case the lumped mass projection technique is applied again, to obtain a node-based current density. In the terminology of finite-difference/finite-volume methods, this would roughly amount to computing the density relative the dual volumes (co-volumes) of the mesh. Namely,

$$\rho_A^N = \frac{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A \, \rho_0(\boldsymbol{X})/J \; dV \right)}{\left(\mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{V_e} N_A \; dV \right)} \tag{8.6}$$

Remark 20 *Also the post-processing approach for the density conserves the total mass.*

8.4 CFL condition

First, it is important to stress that the Courant-Friedrichs-Levy (CFL) condition *must* incorporate the effects of the artificial viscosity, otherwise the code can generate so-called "q-instabilities". The proposed algorithm, when only one iteration of the predictor/multi-corrector approach is applied, yields exactly the same discrete equations generated by the first-order space-time method described in [43]. Since the first iteration is the most restrictive, it is sufficient to derive the CFL condition based on this first-order method. The von Neumann stability analysis developed in [43] for a linear one-dimensional advection-diffusion equation, yields:

$$2 \frac{\Delta t}{\Delta h_e^2} (\nu_{art_e} + c_{s_e}^2 \tau) \leq CFL < 1 \tag{8.7}$$

where, with respect to the e-th element, c_{s_e} is the advective velocity (the speed of sound in our case), ν_{art_e} is the artificial viscosity, Δh_e is the mesh characteristic length (the minimal distance between element nodes), and τ is the stabilization time-scale parameter.

Recalling the expression for the stabilization tensor (4.10),

$$\boldsymbol{\tau} = \tau \boldsymbol{A}_0^{-1} = \frac{\Delta t}{2\, CFL} \boldsymbol{A}_0^{-1} \tag{8.8}$$

it is easily seen that the Jacobian $\boldsymbol{A}_0^{-1}$ plays simply the role of a scaling term, making the equations dimensionally consistent. Substituting $\tau = \Delta t / (2\, CFL)$ in (8.7), and rearranging like terms, we obtain:

$$0 = c_{s_e}^2 \, \Delta t^2 + 2 \nu_{art_e} \, CFL \, \Delta t - (CFL)^2 \, \Delta h_e^2 \tag{8.9}$$

$$1 > CFL \tag{8.10}$$

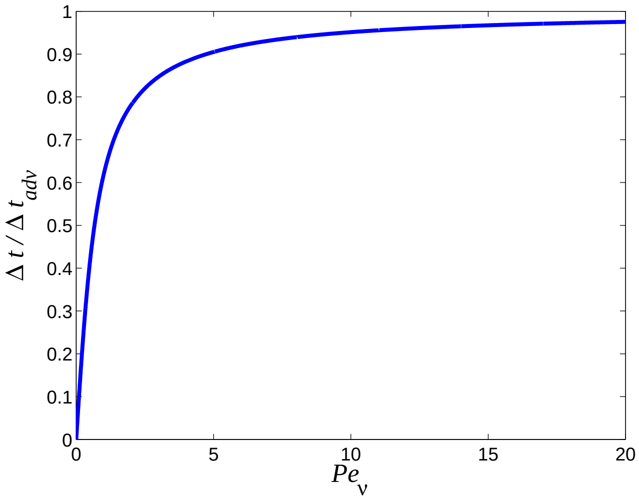

Figure 8.2. Behavior of $\Delta t/\Delta t_{adv}$ as a function of Pe_ν. The optimum is to have $\Delta t/\Delta t_{adv} = 1$, that is, the CFL condition is uniquely due to the speed of sound. This situation is achieved in the limit $Pe_\nu \to \infty$.

(8.9) is a quadratic equation that can be easily solved for $\Delta t > 0$:

$$\begin{aligned}\Delta t &= CFL\frac{\sqrt{\nu_{arte}^2 + \Delta h_e^2\ c_{s_e}^2} - \nu_{arte}}{c_{s_e}^2} \\ &= CFL\frac{\Delta h_e}{c_{s_e}}\left(\sqrt{\frac{1}{4Pe_\nu^2}+1} - \frac{1}{2Pe_\nu}\right) \\ &= \frac{\Delta t_{adv}}{2Pe_\nu}\left(\sqrt{1+4Pe_\nu^2} - 1\right)\end{aligned} \tag{8.11}$$

where $Pe_\nu = c_{s_e}h_e/(2\nu_{arte})$, $\Delta t_{adv} = CFL\Delta h_e/c_{s_e}$, and Δh_e is the minimum of the distances (measured in the current configuration) between two distinct vertices of the e-th element. A plot of $\Delta t/\Delta t_{adv}$ is presented in Figure 8.2. Let us now analyze the limit behavior of (8.11) holding the speed of sound c_{s_e} fixed and varying ν_{arte}:

$$\Delta t \sim CFL\frac{\Delta h_e}{c_{s_e}} = \Delta t_{adv}, \qquad \text{as } \nu_{arte} \to 0 \tag{8.12}$$

$$\Delta t \sim CFL\frac{\Delta h_e^2}{2\nu_{arte}} = \Delta t_{\nu_{arte}}, \qquad \text{as } \nu_{arte} \to \infty \tag{8.13}$$

It is clear from this analysis that high values of the artificial viscosity have a detrimental effect on the time step magnitude. However, the values of ν_{arte} were usually of the same order of (and typically smaller than) the product $\Delta h_e\ c_{s_e}$, so that the additional constraint due to the artificial viscosity is only 10–20% more severe than the purely advective CFL condition, since usually $Pe_\nu \in [3, 10]$.

Chapter 9

One-dimensional tests

A number of one-dimensional Riemann problems, for which exact solutions are easy to compute, was solved numerically (the reader can refer to the book by Toro [50] on Eulerian numerical methods for a comprehensive description). The SUPG method was compared with a standard hydrocode implementing a HEMP-viscosity without limiter (see, e.g., Benson [4], and [41] for complete details). The HEMP artificial viscosity had constants equal to 1.5 for the von Neumann-Richtmyer part and 0.06 for the linear part. Results for the planar version of the implosion test devised by Noh [39] are also presented. Table 9.1 summarizes the initial conditions and Table 9.2 presents the exact intermediate $*$-state values at the final time of each simulation. All one-dimensional simulations were run at $CFL = 0.9$, for both the standard hydrocode and the SUPG method.

Test	$\rho_0^{(L)}$	$\rho_0^{(R)}$	$p_0^{(L)}$	$p_0^{(R)}$	$v_0^{(L)}$	$v_0^{(R)}$	γ
Sod	1.0	0.125	1.0	0.1	0.0	0.0	1.4
LWC	1.0	1.0	1000	0.01	0.0	0.0	1.4
2SH	5.99924	5.99242	460.894	46.0950	19.5975	-6.19633	1.4
Noh	–	1.0	–	0.0	–	-1.0	5/3

Table 9.1. One-dimensional test suite: Initial conditions. (L) and (R) stand for the left and right states, respectively. The following nomenclature is used: "Sod" refers to the Sod test, "LCW" refers to the left half of the Woodward-Colella blast test problem, "2SH" refers to the two-shock problem, and "Noh" refers to the Noh test. In the case of the Noh test some entries of the table are missing, since there is no left state, but just a rigid wall boundary condition.

Test	$\rho_*^{(L)}$	$\rho_*^{(R)}$	p_*	v_*
Sod	0.42632	0.26557	0.30313	0.92745
LWC	0.57506	5.99924	460.894	19.5975
2SH	14.2823	31.0426	1691.64	8.68975
Noh	4.0	4.0	4/3	0.0

Table 9.2. One-dimensional test suite: Exact solution of the Riemann problem ($*$-states). The nomenclature for the test cases is the same as in Table 9.1. Notice that in the case of the Noh test no contact discontinuity is generated ($\rho_*^{(L)} = \rho_*^{(R)}$).

9.1 Sod's problem

The Sod's test [45] is presented in Figures 9.1 and 9.2. The hydrocode delivers good performance, but a pronounced overshoot is present in the energy plot and the velocity shows low accuracy in the representation of the solution past the shock front. The results for the SUPG method are better for the velocity. Although milder, an overshoot is still present for the energy. One plausible explanation is the fact that, in contrast to the density, the energy is just obtained by a point-wise calculation at the nodal points. A conservative L^2-projection technique could be more effective. The contact discontinuity is captured within 2–3 elements by the SUPG method, due to the specific pre-processing of the initial condition for the density, as mentioned in section 8.2.

9.2 Left-half of Woodward-Colella blast

The left half of the Woodward-Colella [54, 53] interacting blast wave is shown in Figures 9.3 and 9.4. Wiggles are clearly visible behind the shock location for velocity, pressure, and density in the case of the hydrocode, while they are absent in the case of the SUPG method. In addition, the pronounced over/under-shoot in the hydrocode solution for the velocity/pressure at the beginning of the expansion (about $x = -.25$) is attenuated in the SUPG plots.

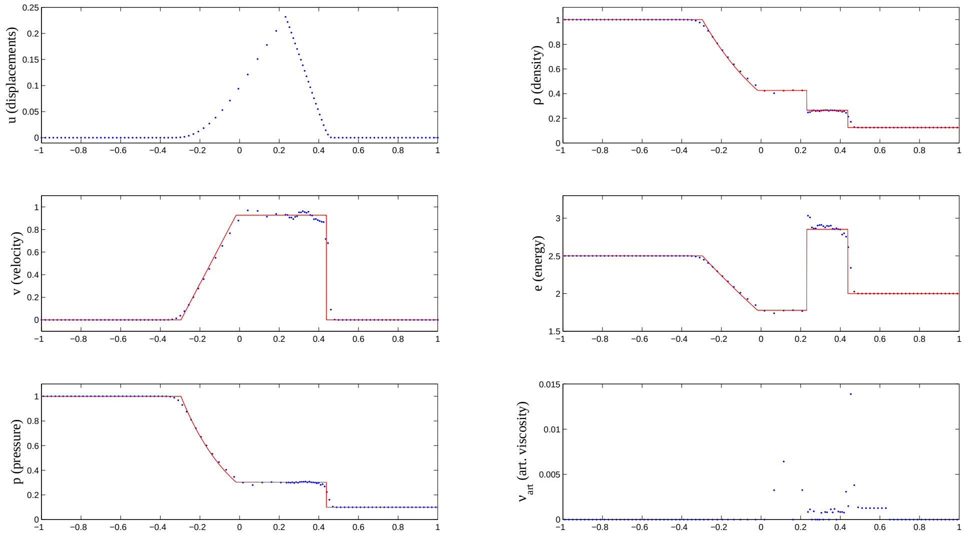

Figure 9.1. Hydrocode solution for the Sod test. The exact solution is represented by the continuous line.

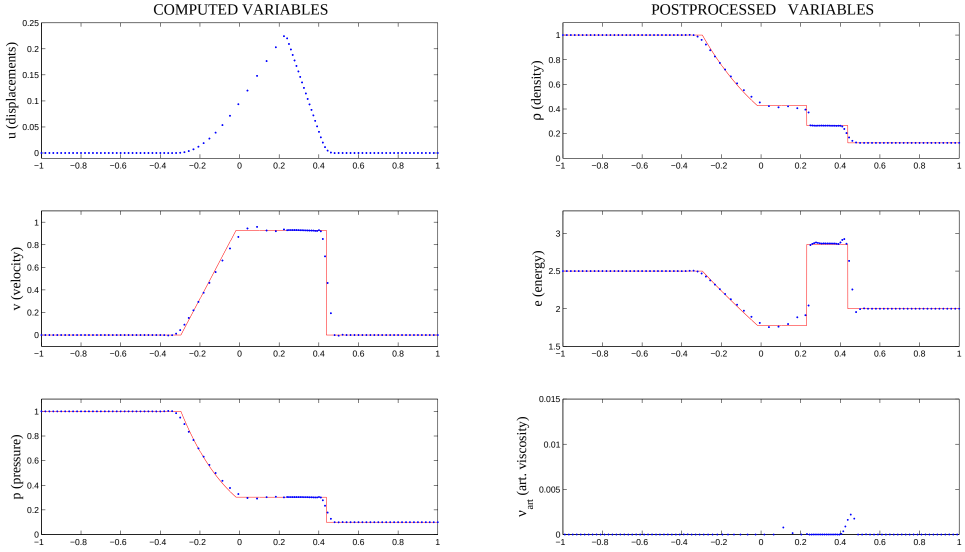

Figure 9.2. SUPG solution for the Sod tests. The exact solution is represented by the continuous line.

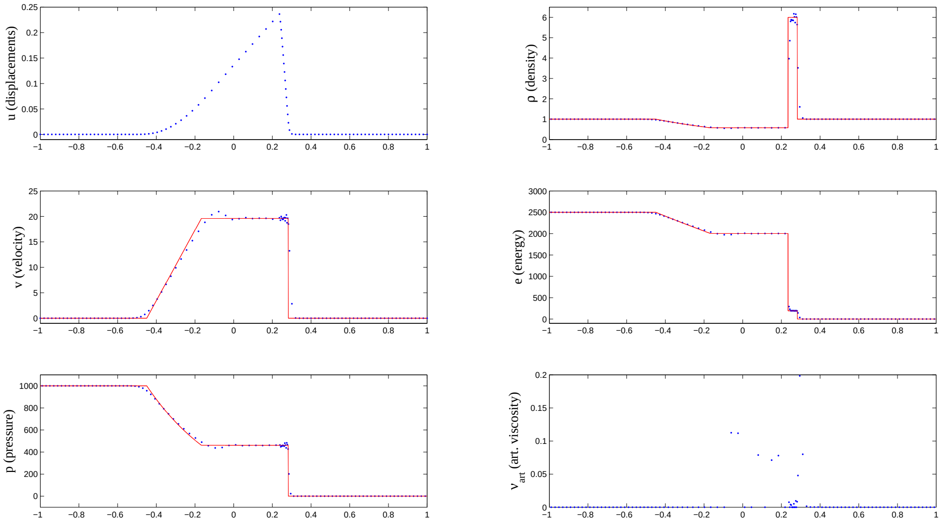

Figure 9.3. Hydrocode solution for the left-half of the Woodward-Colella blast test. The exact solution is represented by the continuous line.

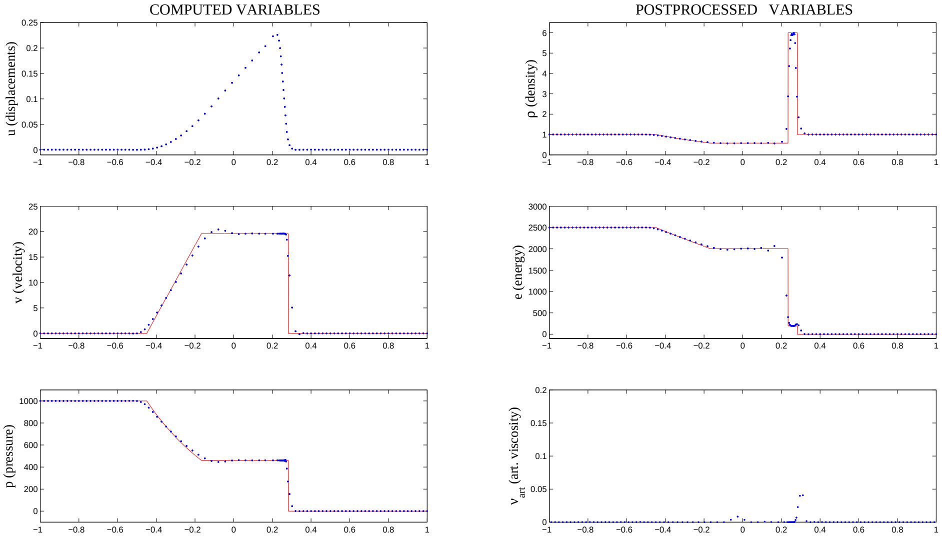

Figure 9.4. SUPG solution for the left-half of the Woodward-Colella blast test. The exact solution is represented by the continuous line.

9.3 Two-shock problem

The two-shock test [50], shown in Figures 9.5 and 9.6, is the most demanding of the suite in terms of robustness: A contact discontinuity is generated by the interaction of two strong shocks. The test presents features very similar to implosion calculations. The strong compression undergone by the initial computational domain, spanning the interval $[-1, 1]$, is clearly noticeable.

The hydrocode solution suffers from a few wiggles in the velocity and pressure plots, absent in the case of the SUPG method. Although over/under-shoots are present in the SUPG solution for the energy and density, they tend to be attenuated, compared to the hydrocode results.

Remark 21 *The Noh-type artificial heat flux is usually deprecated because it can smear contact discontinuities. In our experience, however, contact discontinuities are usually captured by the SUPG method within* 2–4 *elements, this spreading being mainly due to the pre-processing of the initial density. After monitoring the evolution of the contact layers throughout the simulations, no increase in their width was observed as time progressed. A plausible explanation is that the artificial heat flux adopted herein includes* only *a quadratic term in the mesh scaling, while the general form of the Noh correction includes also a linear term. It seems that even if mild oscillations in the velocity are switching on the artificial heat flux at the contact, its scaling keeps it within the numerical error threshold.*

9.4 Planar Noh's test

In the planar Noh test [39] – a robustness test – a bar of perfect gas is driven against a hard wall (the node at the left end of the computational domain), where zero velocity/displacement boundary conditions are imposed. The initial pressure of the gas is set to zero (namely 10^{-14}), to obtain an infinite strength shock.

The hydrocode delivers good performance although a few wiggles are present past the shock front for pressure and velocity. There is wall *overheating* in the element facing the left boundary (a noticeable spike in the energy and a severe dip in the density, see Fig. 9.7).

The SUPG method (see Fig. 9.8) shows no wiggles for pressure and velocity and some moderate *underheating* at the wall, due to the use of the Noh-type heat flux correction.

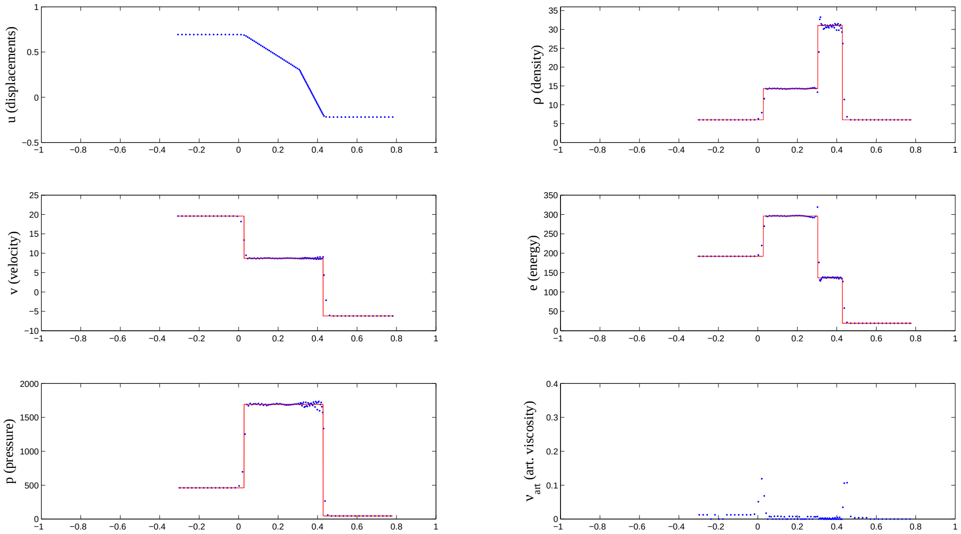

Figure 9.5. Hydrocode solution for the two-shock problem. The exact solution is represented by the continuous line.

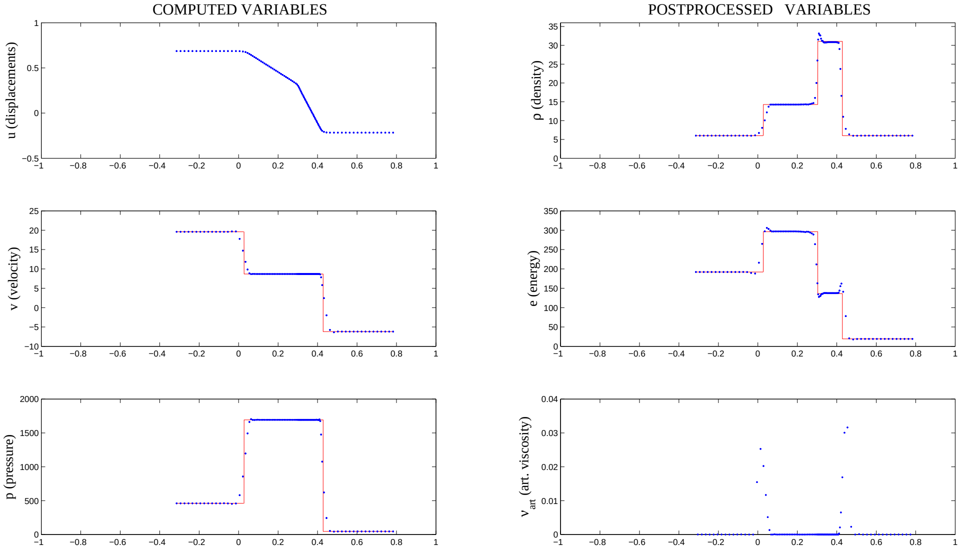

Figure 9.6. SUPG solution for the two-shock problem. The exact solution is represented by the continuous line.

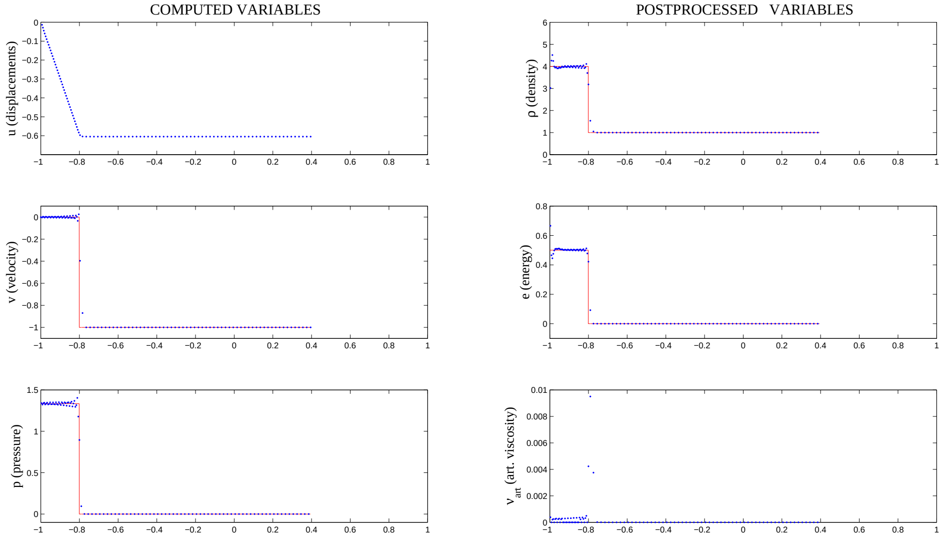

Figure 9.7. Hydrocode solution for the one-dimensional, planar Noh test. The exact solution is represented by the continuous line.

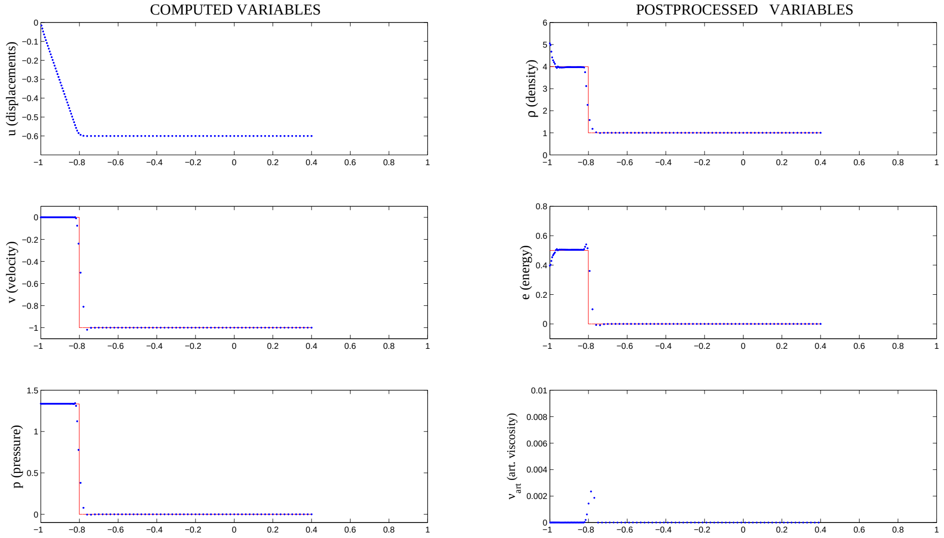

Figure 9.8. Second-order Hydro-SUPG for the one-dimensional, planar Noh test. The exact solution is represented by the continuous line.

9.5 Overall assessment for the one-dimensional tests

To comment on the overall performance of the SUPG method with respect to the standard hydrocode approach, it is fair to say that usually the SUPG method outperforms the hydrocode in terms of the displacement, velocity and pressure solution, it is comparable in terms of the density solution and slightly worse for the internal energy, which is however post-processed in a very crude way.

Chapter 10

Two-dimensional numerical tests

A number of classical tests, very well documented in the technical literature (see, e.g. [3, 4, 9, 8, 7] and references therein), is presented. Computations were performed on isoparametric Q1-quadrilateral and P1-triangular elements. Since the mesh topology represents an integral part of the tests, and, at the same time, very little work has been published in the context of triangular/tetrahedral meshes, our approach has been to design variants of the standard tests by subdividing each quadrilateral in the original meshes into two triangles. The choice of the splitting is not unique in general and, in all instances, results from multiple strategies are compared.

10.1 Preliminary test on triangular meshes: Piston problem

A first issue to be addressed in detail is whether or not the current formulation generates artificial stiffness or even locking on triangular meshes. Two preliminary tests were performed by meshing the domain $[0, 1] \times [0, 0.1]$ (see Fig. 10.1) with 500 elements of 1:1 aspect ratio, and 50 elements with 1:20 aspect ratio. Both simulations were run at $CFL = 0.75$. These apparently trivial meshes are usually adopted to test whether a finite element formulation induces *locking* in the incompressible limit. The right boundary acts as a piston moving to the left with unit velocity, and generates a shock wave. Unit density and zero internal energy (namely 10^{-14}) initial conditions are imposed.

In the present case, the equations allow compressibility effects, so locking is unlikely, but artificial stiffness may occur. From the results shown in Figure 10.2, no locking or artificial stiffness occurs, and the pressure profiles are comparable in quality with simulations performed on quadrilateral elements.

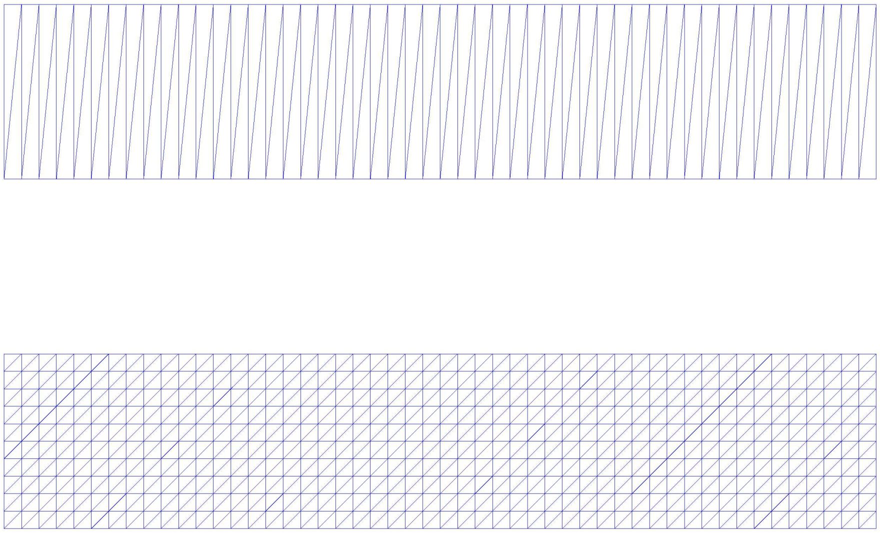

Figure 10.1. Piston problem for triangles: Initial meshes.

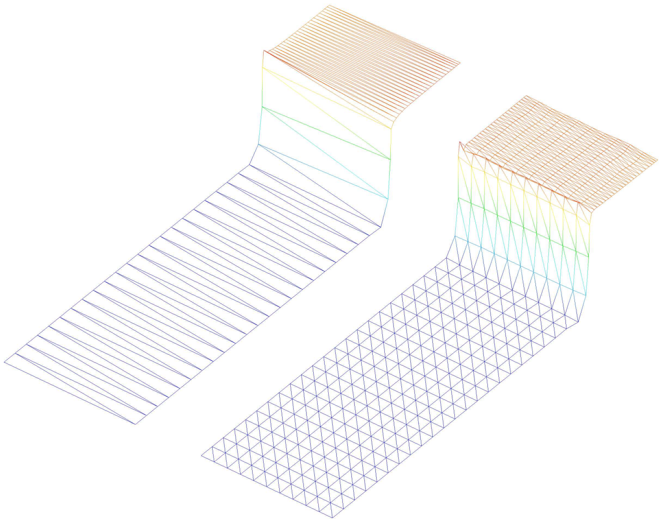

Figure 10.2. Piston problem for triangles: Pressure. The two solutions are evaluated at the same time step. Notice the absence of any node-to-node oscillations, typical of artificially stiff P1 elements.

10.2 Saltzmann test in cartesian coordinates

The Saltzmann test evaluates the ability of a distorted mesh to capture the features of a planar shock. A rectangular domain of gas is initially at rest. The bottom boundary moves with unit velocity and generates a compression shock propagating upwards through the domain. All other boundary conditions are of "roller" type, that is, zero normal velocity (and, consequently, zero normal displacement). The Saltzmann test is both a robustness and accuracy test. The initial meshes, an integral part of the test case, are presented in Figure 10.3.

Three variants, differing by a scaling transformation, are presented. All meshes are generated from a common layout of 10 quadrilateral elements in the horizontal direction and 100 quadrilateral elements in the vertical direction, for a total of 1111 nodes. Nodes are located according to the following scheme, where x is the horizontal coordinate and y is the vertical coordinate:

$$x_{ij} = \alpha_x \Delta x \ (i-1) \tag{10.1}$$

$$y_{ij} = \alpha_y \left(\Delta y \ (j-1) + \Delta x (11-i) \sin\left(\frac{\pi (j-i)}{100} \right) \right) \tag{10.2}$$

with $i = 1, 2, \ldots 11$ and $j = 1, 2, \ldots 101$, $\Delta x = \Delta y = 0.01$, and α_x and α_y are scaling factors. Triangular meshes were generated by splitting each quadrilateral into two triangles. The three proposed variants are the following:

1. The 1:1 aspect ratio test (i.e., $\alpha_x = 1$ and $\alpha_y = 1$) on the rectangular domain $[0,1] \times [0,0.1]$ is the standard Saltzmann test.
2. The 1:2 aspect ratio test (i.e., $\alpha_x = 0.5$ and $\alpha_y = 1$) on the domain $[0,1] \times [0,0.05]$, was orignally proposed by Campbell and Shaskov [7], to evaluate the performance of hydrocodes when anisotropies are present in the mesh.
3. The 100:1 aspect ratio test (i.e., $\alpha_x = 100$ and $\alpha_y = 1$) on the domain $[0,1] \times [0,10]$ was originally proposed by Margolin [37], to evaluate how very high aspect ratio elements would affect the accuracy of hydrocode simulations.

For each of the three variants, a standard quadrilateral grid (mesh (I) in Fig. 10.3) and two triangular grids (mesh (II) and (III) in the same figure) were adopted. It is important to realize that although generated by splitting each of the quadrilaterals of mesh (I) into two triangles, mesh (II) and (III) are very different in quality. In fact, *none* of the triangular elements of mesh (II) is obtuse while *all* of the triangles of mesh (III) are. Therefore, the quality of mesh (III) is significantly poorer than mesh (II), although the location of the nodes is the same. In particular, an automatic mesh generator would *most likely* connect nodes according to mesh (II) rather than mesh

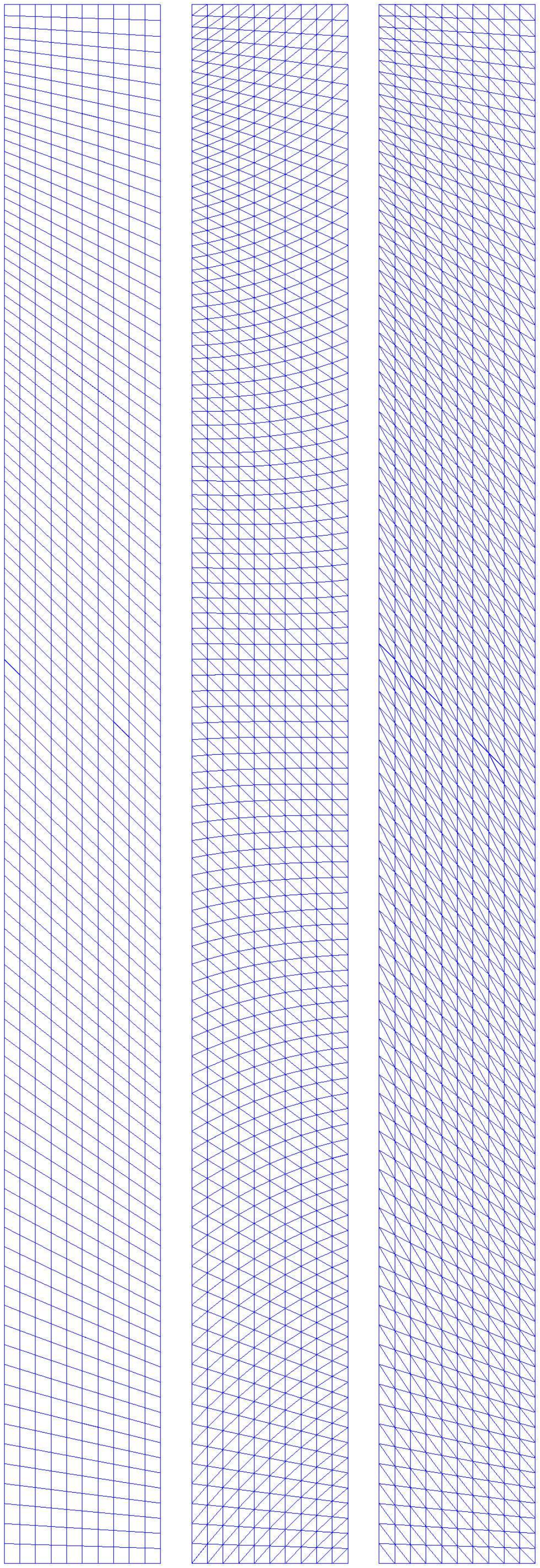

Figure 10.3. Initial mesh for the Saltzmann 1:1 test. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). The 1:2 and 100:1 meshes are obtained by appropriate scaling along the horizontal axis.

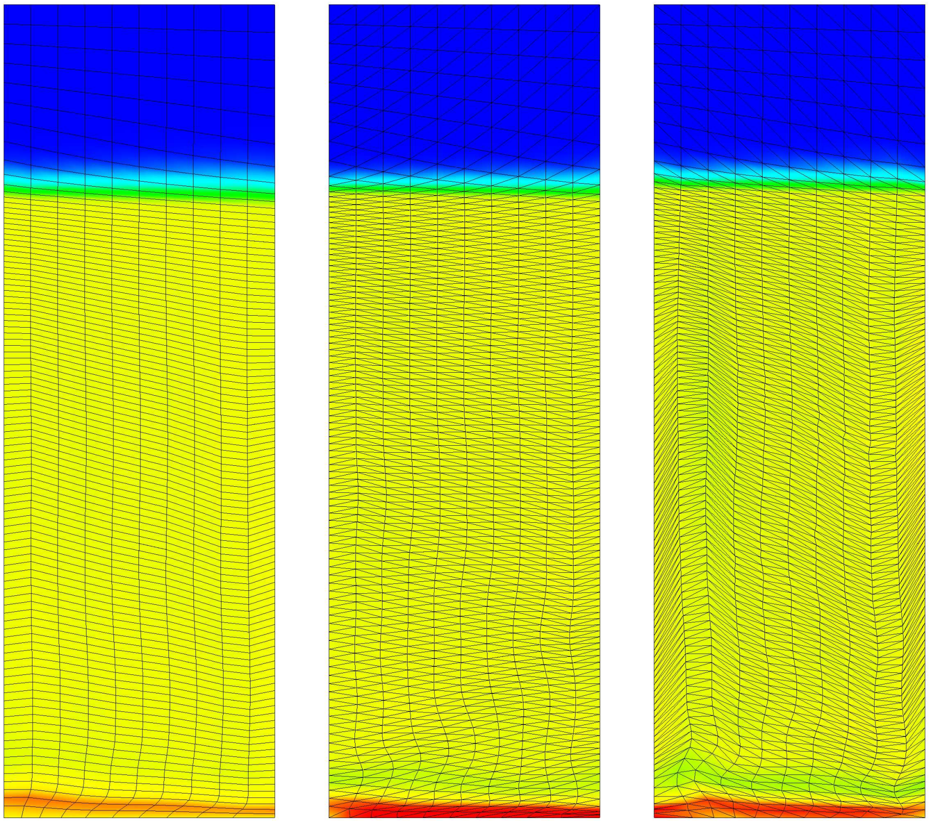

Figure 10.4. 1:1 Saltzmann solution at $T = 0.7$. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)).

(III), since it usually incorporates tools to assess the quality of elements. Notice also that the 1:2 variant exacerbates the difference between the meshes, while the 100:1 tends to level it.

All simulations were performed at $CFL = 0.75$.

Figures 10.4, 10.5, and 10.6 show the post-processed density solutions for the 1:1, 1:2, and 100:1 test, respectively, at time $T = 0.7$, shortly before the shock generated by the piston reflects off the upper boundary. In all three tests, the density profiles for mesh (I) and (II) look smooth and the grids do not present strong distortions, with the exception of the region near the piston, where some under-heating (an overshoot of the density corresponding to an undershoot in energy) is taking place. The density solution and grid deformation are smooth only in the 1:1 and 100:1 tests for mesh (III). Mesh (III) undergoes intense distortion in the 1:2 test. A simple explanation is that mesh (III) is intrinsically much poorer than mesh (II), as previously mentioned. It is quite surprising, however, that the degradation of the performance of the method in the case of mesh (III) is only significant in the 1:2 case.

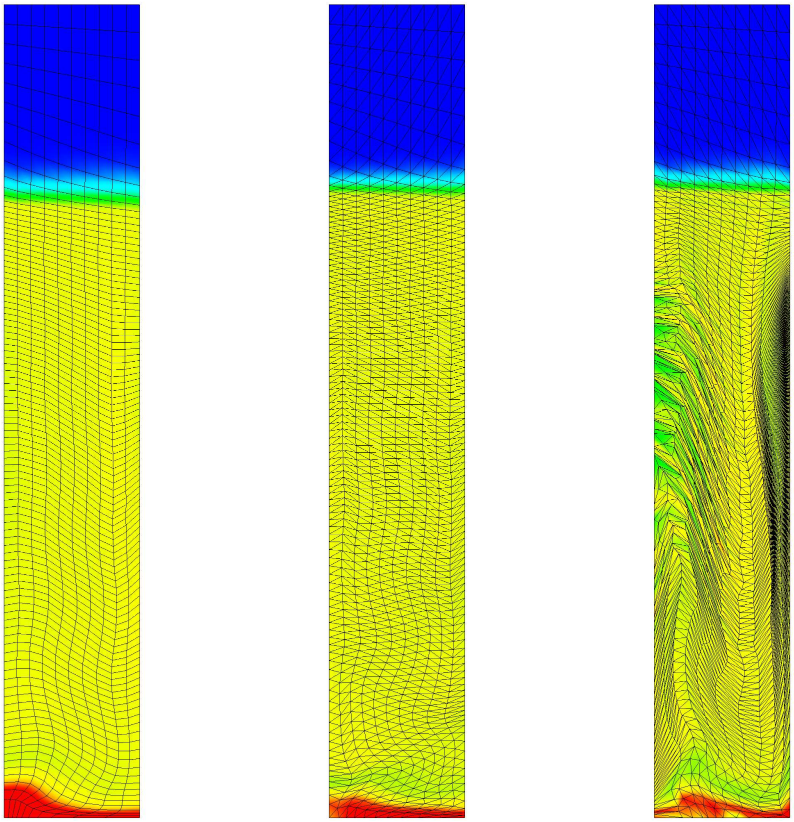

Figure 10.5. 1:2 Saltzmann solution at $T = 0.7$. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)).

Quantitative results are shown in Figures 10.7, 10.8, and 10.9. For the 1:1 and 1:2 tests (Fig's. 10.7 and 10.8, resp.), the under-heating is more pronounced, and lack of perfect planar symmetry is observable in the normalized plot for the horizontal velocity component v_1, which is instead much smaller in the 100:1 test. Overall, for all three variants there is good agreement with state-of-the-art computations [8, 7]. As mentioned before, results for mesh (III) are clearly the poorest in the 1:2 case, while for the 1:1 and 1:100 tests, although inferior, they compare well with meshes (I) and (II).

Remark 22 *It is important to notice that the simulations on triangles are essentially of the same accuracy as the simulations on quadrilaterals, whenever the quality of the meshes is comparable.*

Remark 23 *The 100:1 test does not pose any problematic issue for the current algorithm, contrary to the majority of hydrocodes. In fact, the 100:1 test has the best results compared with the 1:1 and 1:2 test.*

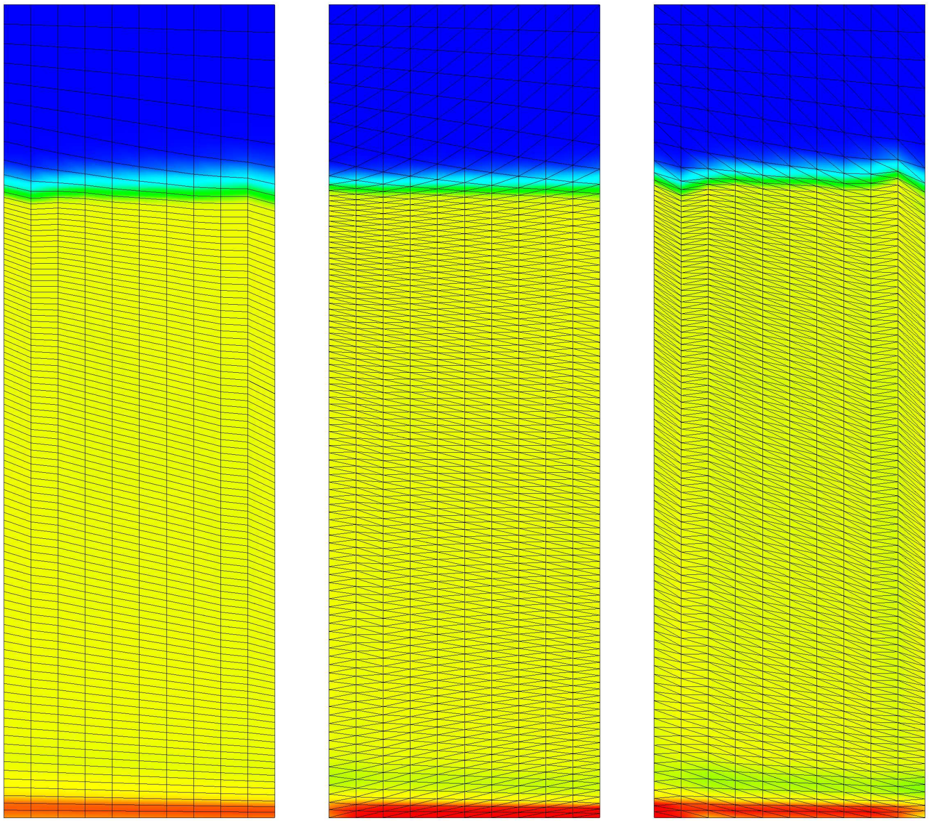

Figure 10.6. 100:1 Saltzmann solution at $T = 0.7$. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). The horizontal axis has been rescaled by a factor 0.01, for presentation purposes.

Remark 24 *It is clearly seen on the top row of Figure 10.8 that the undershoot in the pressure causes negative values of the pressure. Such unphysical undershoots are limited to* $0.1 - 2.0\%$ *for the triangles, and are even smaller for the quadrilaterals. The explanation for the good performance of the code in the presence of small negative pressures has to do with the fact that in Lagrangian coordinates, a negative pressure* does not necessarily *lead to a negative density, since the density is computed using the equation* $\rho J = \rho_0$*, and no volume inversions occurred in the computations.*

In principle, there is no guarantee for positivity of the pressure in the presence of large errors in the velocities. This is an important issue and is currently under investigation. However, the typical pathologies related to spurious negative values for the pressures, such as lack of conservation, large errors in the shock wave arrival times, and large errors in the plateaus of the solution past a shock, have not manifested themselves in the many simulations we performed. More work will be devoted to at least mitigate and ideally eliminate this problem, especially in connection to complex equations of state, which are typically used in tabulated form.

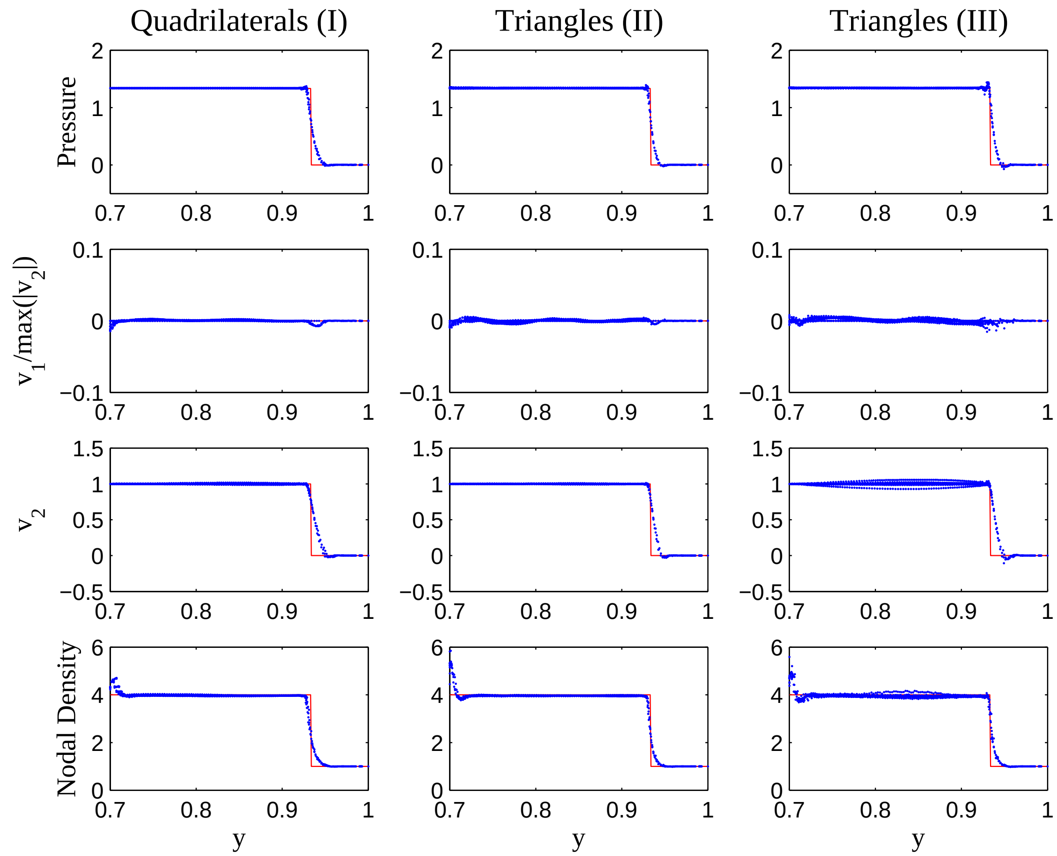

Figure 10.7. Saltzmann 1:1 test. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). Along the rows from top to bottom are plotted – as a function of the vertical coordinate y – pressure, normalized horizontal velocity ($v_1/\max(|v_2|)$), vertical velocity v_2, and nodal density. The exact solution is represented by the continuous red line, the dots represent *all* the nodal values of the numerical solution (the x-coordinate locations are compressed into a single plane).

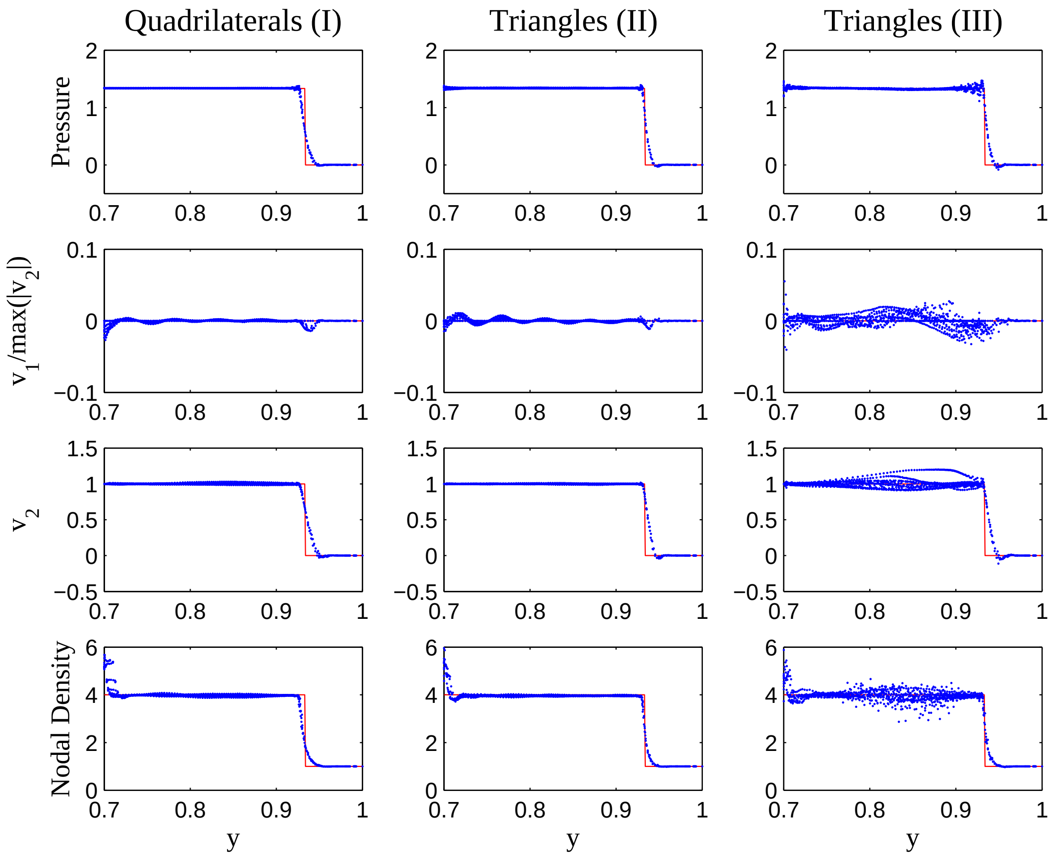

Figure 10.8. Saltzmann 1:2 test. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). Along the rows from top to bottom are plotted – as a function of the vertical coordinate y – pressure, normalized horizontal velocity ($v_1/\max(|v_2|)$), vertical velocity v_2, and nodal density. The exact solution is represented by the continuous red line, the dots represent *all* the nodal values of the numerical solution (the x-coordinate locations are compressed into a single plane).

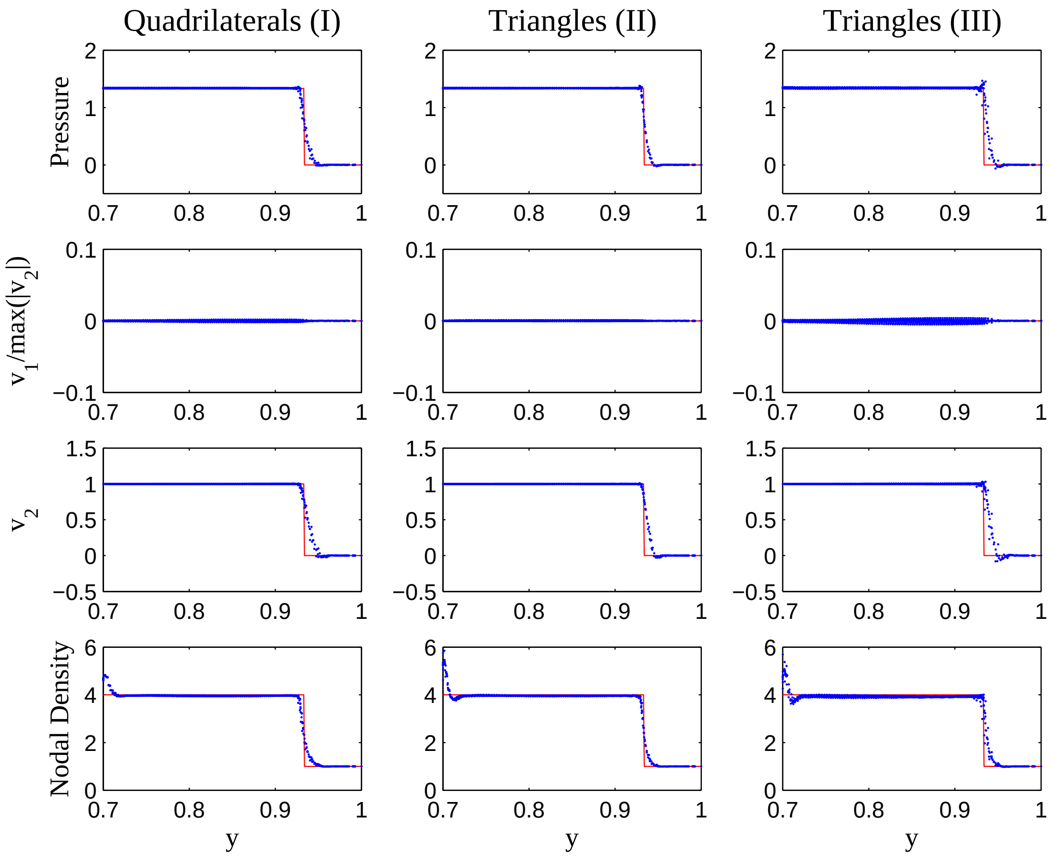

Figure 10.9. Saltzmann 100:1 test. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). Along the rows from top to bottom are plotted – as a function of the vertical coordinate y – pressure, normalized horizontal velocity ($v_1/\max(|v_2|)$), vertical velocity v_2, and nodal density. The exact solution is represented by the continuous red line, the dots represent *all* the nodal values of the numerical solution (the x-coordinate locations are compressed into a single plane).

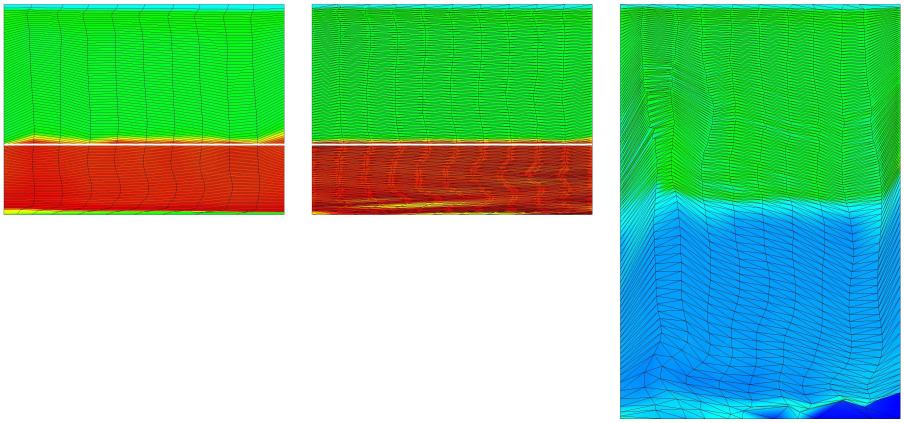

Figure 10.10. Saltzmann 1:1 test solution (density color plot). Left: Quadrilaterals (mesh (I)) at $T = 0.925$. Center: Triangles (mesh (II)) at $T = 0.925$. Right: Triangles (mesh (III)) at $T \approx 0.85$. The white horizontal marker indicates the location of the shock for the exact solution. Notice that the 1:1 test on mesh (III) could not be run to the end time due to a volume inversion.

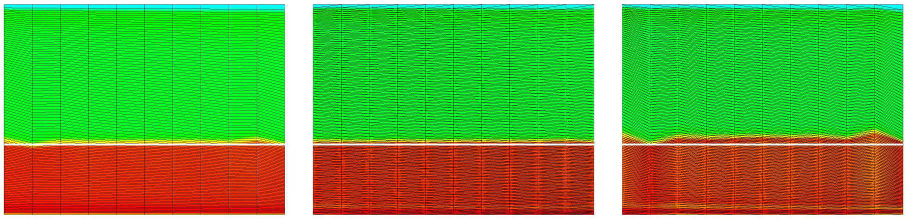

Figure 10.11. Saltzmann 100:1 test solution (density color plot) at $T = 0.925$. Left: Quadrilaterals (mesh (I)). Center: Triangles (mesh (II)). Right: Triangles (mesh (III)). The white horizontal marker indicates the location of the shock for the exact solution.

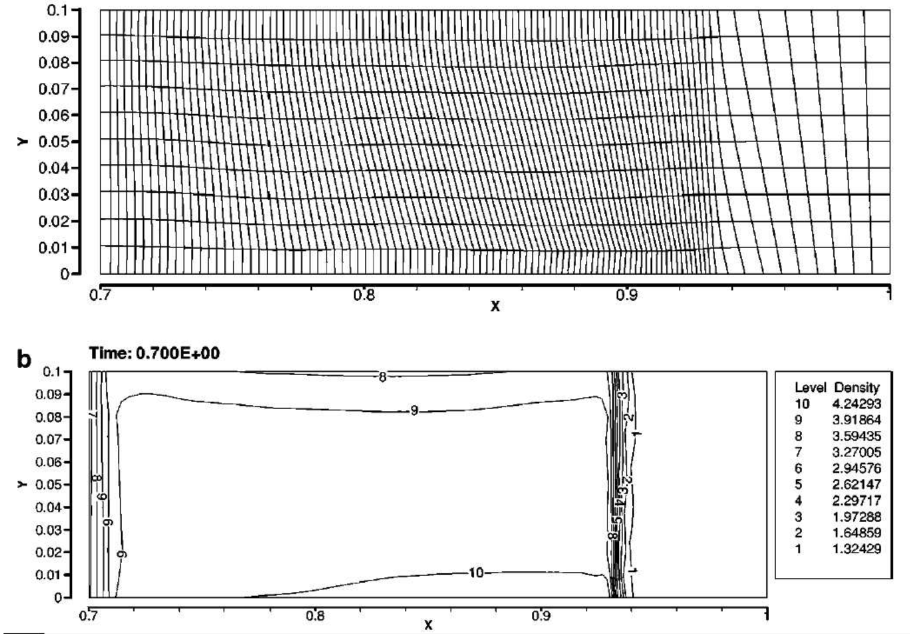

Figure 10.12. Results from [8] on the Saltzmann test.

Figure (10.10) shows the results of a long run 1:1 test. Color plots of the density at $T = 0.925$, are presented for meshes (I) and (II), while the solution for mesh (III) is shown at the earlier time $T \approx 0.85$, when the simulation was arrested by a volume inversion. The white markers in the solutions for meshes (I) and (II) indicate the exact location of the shock generated by the piston, which has reflected twice (the first reflection occurring at the upper boundary, the second at the lower), and is moving upwards. It is easily seen that there is good agreement between the quadrilateral mesh (I) and the triangular mesh (II) on the location of the shock. Figure (10.11) shows the results of the same long run test for the 1:100 aspect ratio meshes. Color plots of the density at $T = 0.925$, are now presented for *all* meshes, since no volume inversions occurred during the simulations. Also in this case, the agreement on the location of the shock for the different meshes is good. The tests shown in Figures 10.10 and 10.11 have the scope of assessing whether mild negativity in the pressures may have very negative consequences on the conservation properties of the algorithm. The good results obtained are reassuring from this perspective.

Overall, the quality of the presented results for the Saltzmann test and its variants is in good agreement with state-of-the-art computations [8, 7] (see, e.g., Fig. 10.12) for quadrilateral elements. The results over triangular meshes maintain the accuracy of the corresponding quadrilateral meshes, as long as the quality of the initial meshes is comparable.

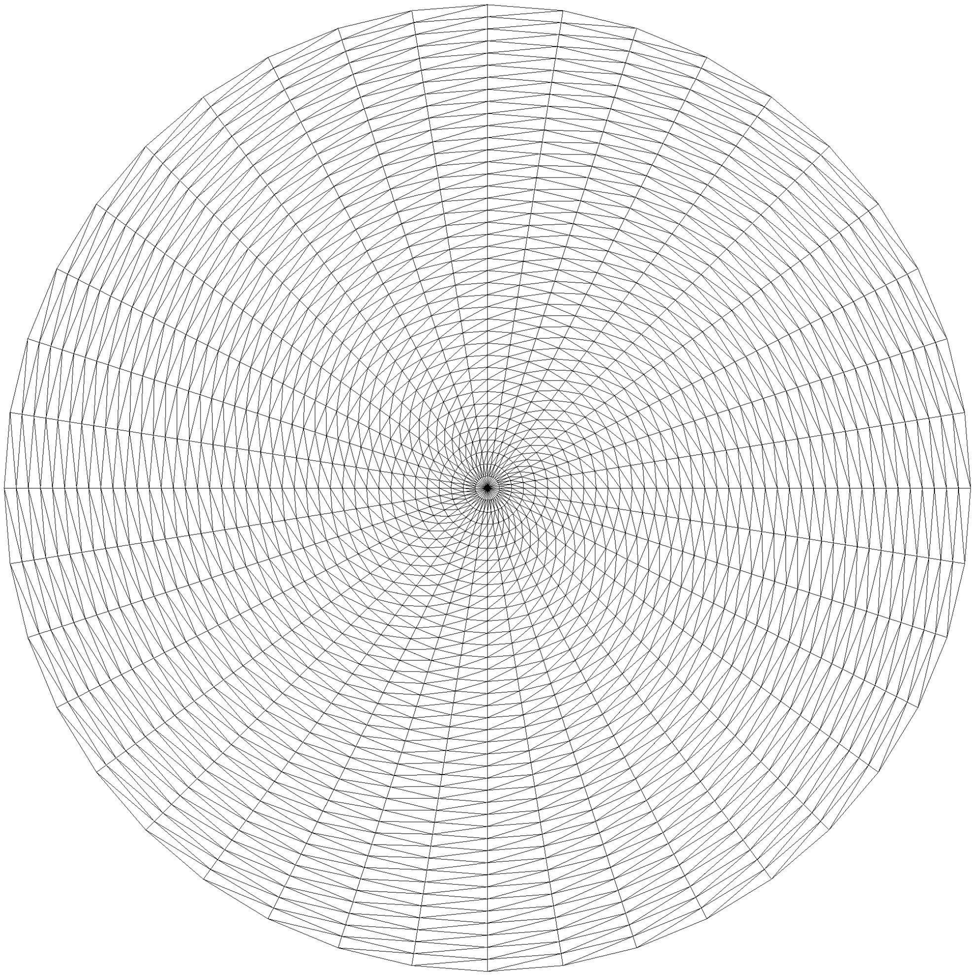

Figure 10.13. Initial configuration for mesh (a), 3,160 triangular elements, 1,601 nodes.

10.3 Two-dimensional Noh test on cartesian grids

10.3.1 Triangular meshes on the unit circle

The Noh [39] test in two dimensions is an implosion test. A cartesian reference frame is used. A number of variants of the test will be considered, to thoroughly evaluate the performance of the SUPG approach. Let us start with three tests on triangular meshes.

The first mesh – indicated as mesh (a) – is obtained by splitting in half (with a directional bias) an underlying 40 × 39 uniformly-spaced quadrilateral mesh. The

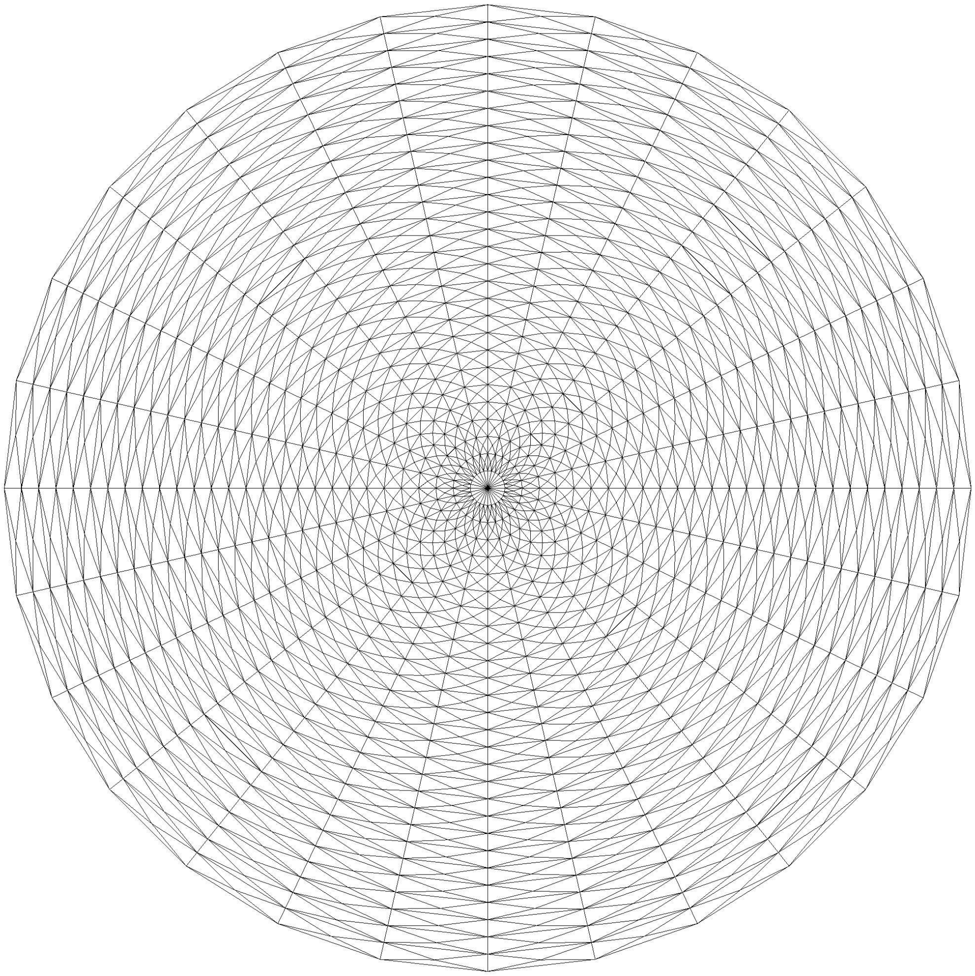

Figure 10.14. Initial configuration for mesh (b), 3,052 triangular elements, 1,541 nodes.

result (Fig. 10.13) is a $3,160$-element mesh with $1,601$ nodes. A second mesh – indicated as mesh (b) – was generated by splitting into four triangles each quadrilateral of a 28×27 uniformly-spaced element mesh. The result is a $3,052$-element mesh with $1,541$ nodes (Fig. 10.14). Mesh (a) and (b) have approximately the same number of nodes and elements, while maintaining different connectivity and geometric characteristics. Finally, mesh (c) is given by a four-fold splitting of an underlying uniformly-spaced quadrilateral mesh of 56×55 elements, yielding a $12,376$-element mesh with $6,217$ nodes. Mesh (c) can be considered as a finer-grained version of mesh (b), and will not be shown for the sake of brevity. Meshes (a) and (b) were originally proposed by Loubère [33], and are characterized by having elements of poor aspect ratio and quality near the origin.

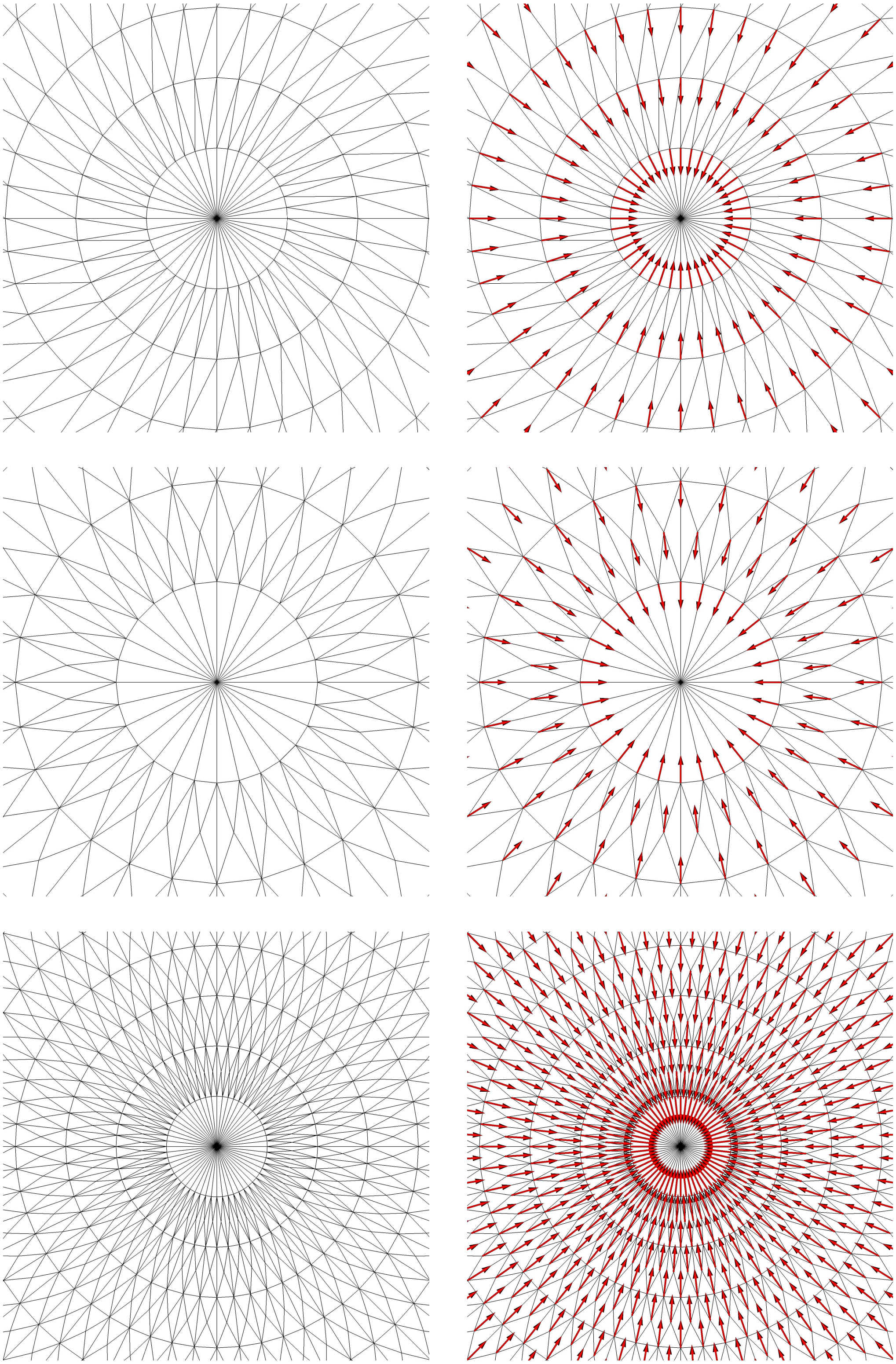

Figure 10.15. Zoom of the initial grid near the origin (left) and of the initial velocity (right). Top: Mesh (a). Center: Mesh (b). Bottom: Mesh (c), 12,376 triangular elements, 6,217 nodes.

The simulations for this first set of three meshes were performed at $CFL = 0.8$. The velocity has an initial uniform radial distribution (all nodal velocity vectors are converging to the origin, and have unit magnitude except the one centered at the origin, which is zero), as shown in Figure 10.15. The initial energy should be zero, but for practical purposes we used the value 10^{-14}. The exact solution for the density behind the shock is 16 and decays as $1+t/r$ in front of the shock, where t is time and $r = \sqrt{x^2 + y^2}$ is the radius. The final time is 0.6, for which the shock, traveling at a velocity 1/3, is found at $r = 0.2$. The numerical results are presented in terms of the final grid configuration and density color plot in Figure 10.16 for mesh (a), Figure 10.17 for mesh (b) and Figure 10.18 for mesh (c). Density elevation plots on the same scale are compared in Figure 10.19. A quantitative comparison is presented in Figure 10.20. Considering the coarseness of meshes (a) and (b), the solutions in terms of pressure, radial velocity and nodal density are satisfactory. Some wall effect is visible near the origin. It is more pronounced in mesh (b) than in mesh (a). For meshes (a) and (b), cylindrical symmetry is reasonably well preserved, in the sense that points at the same radial location but different angular locations maintain such property in the final stage of the simulation (these points overlap in the radial plots of Fig. 10.20). It is important also to realize that the solution for mesh (a) is somewhat inferior, due to a slight rotation of the grid (see the normalized tangential velocity plot). Meshes (a) and (b), although possessing a different geometry, perform almost equivalently in terms of pressure, radial velocity and density. There is very good reliability with respect to changes in mesh topology.

Mesh (c) provides a more accurate solution than meshes (a) and (b): The shock front is sharper and the plateaus of the pressure and density are closer to the exact value. However, very close to the origin, the wall effect is somewhat more intense in amplitude. The reader should notice that convergence is not at risk, since the extent of the region where wall effects are present is actually smaller than in meshes (a) or (b), so that in an L^2- or L^1-sense, the numerical solution is converging to the exact solution. The intensity of the wall effect tends also to break symmetry in that region, a fact that can be explained as follows: Because of the grid-generation strategy for cases (b) and (c), the elements at the origin deteriorate in quality (in terms of skewness and aspect ratio) as the meshes get finer. Mesh (c) is finer than mesh (b), but of poorer quality in a neighborhood of the origin. This may explain the occurrence larger errors, which eventually affect symmetry.

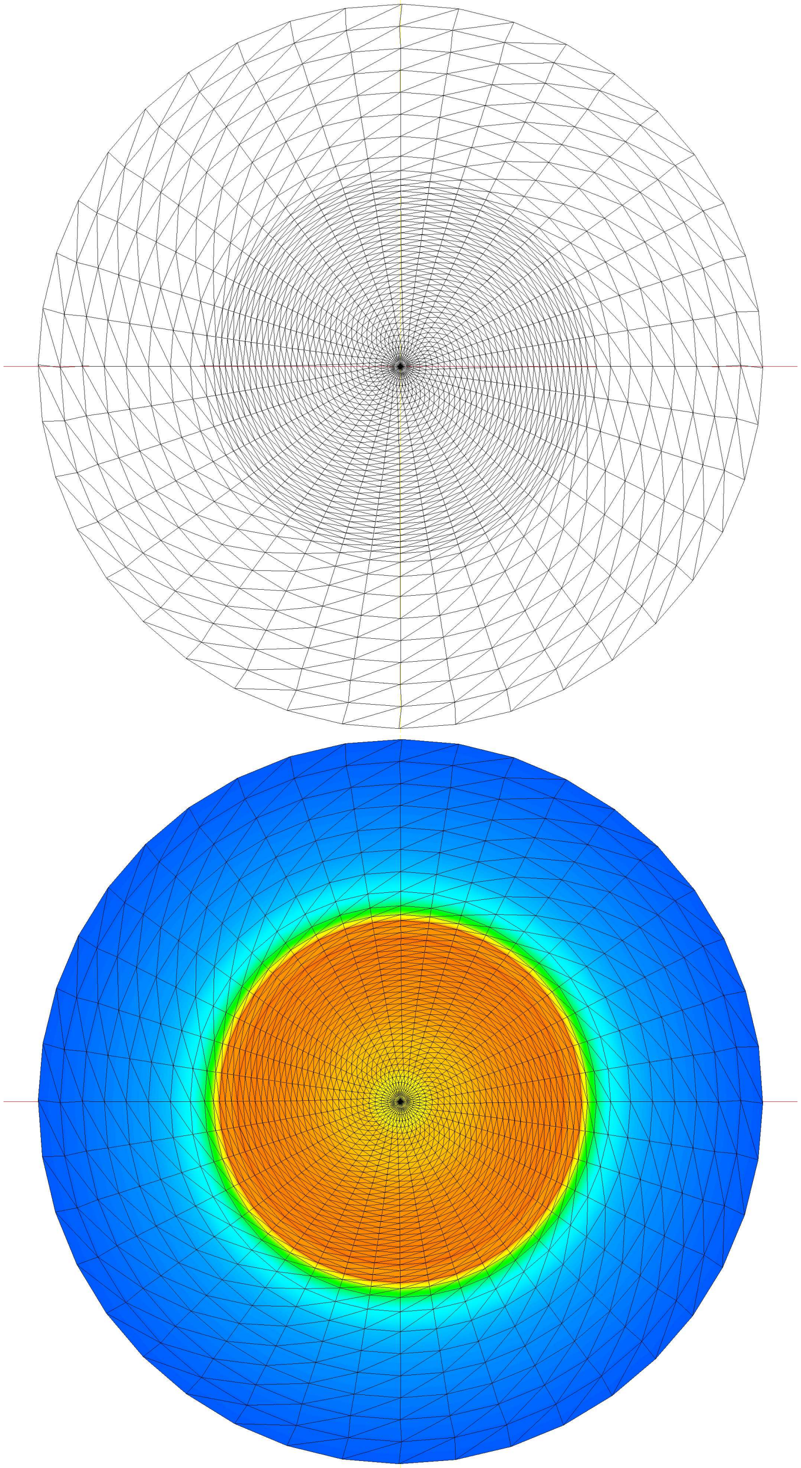

Figure 10.16. Noh test on mesh (a): Final grid (above) and density color plot (below).

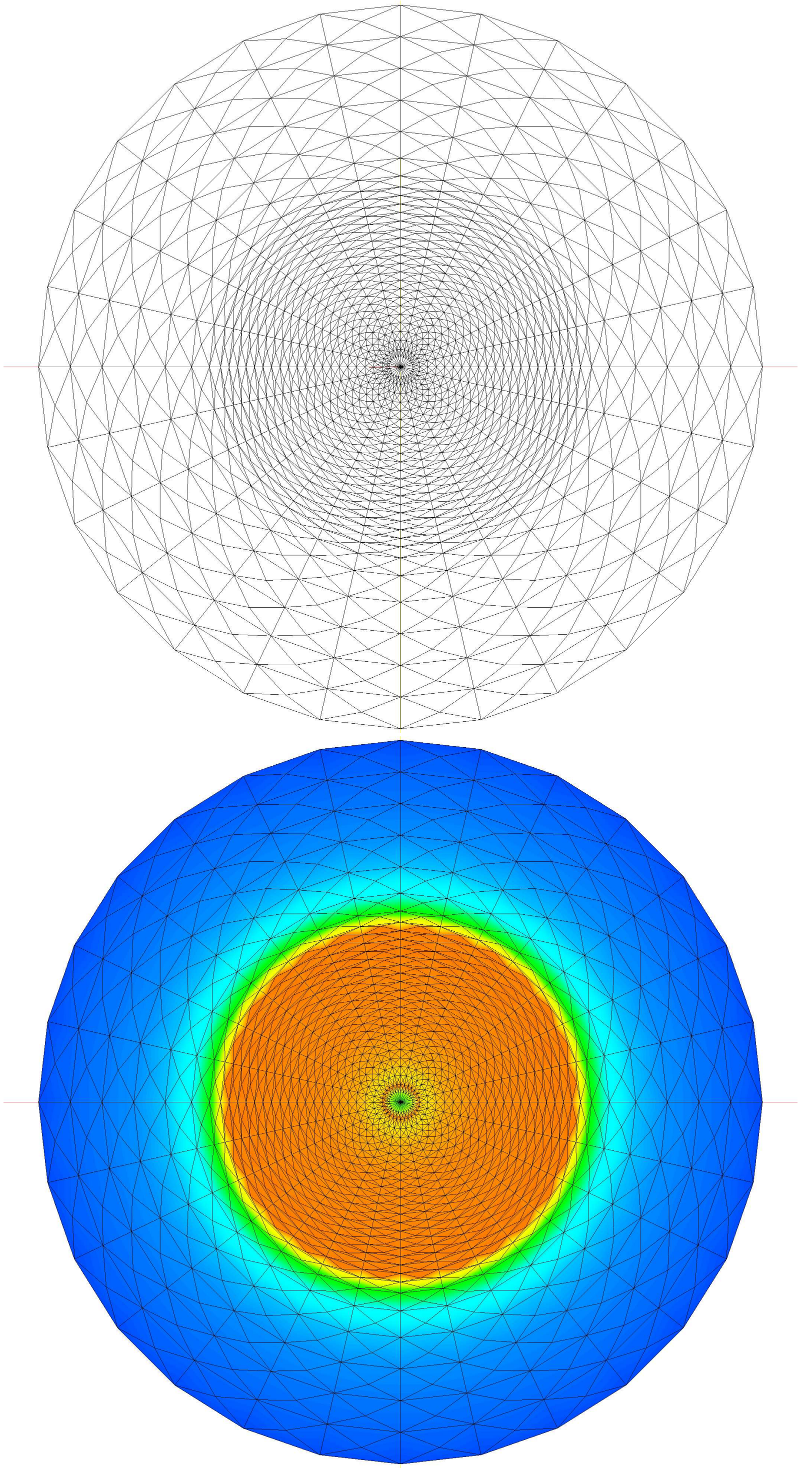

Figure 10.17. Noh test on mesh (b): Final grid (above) and density color plot (below).

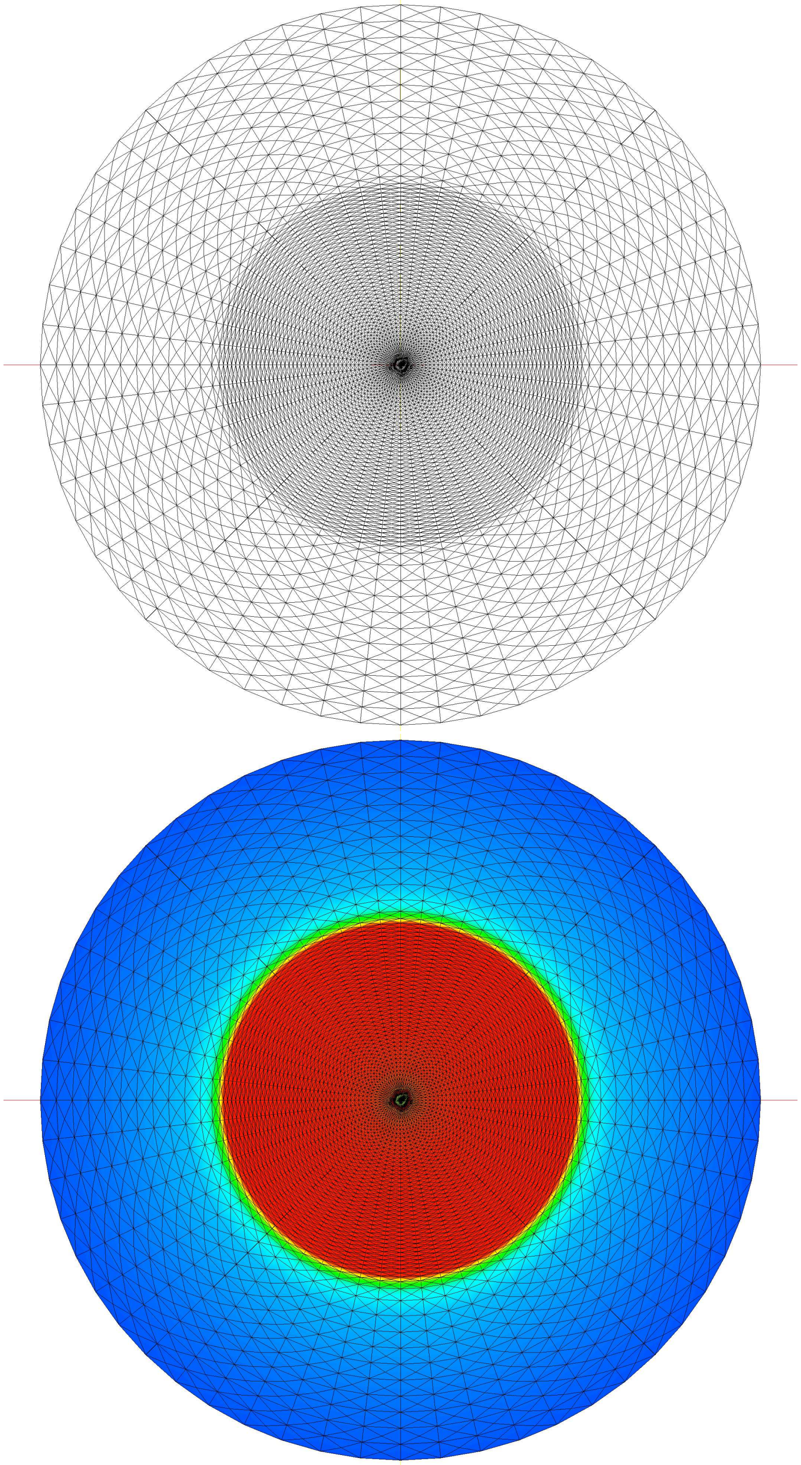

Figure 10.18. Noh test on mesh (c): Final grid (above) and density color plot (below).

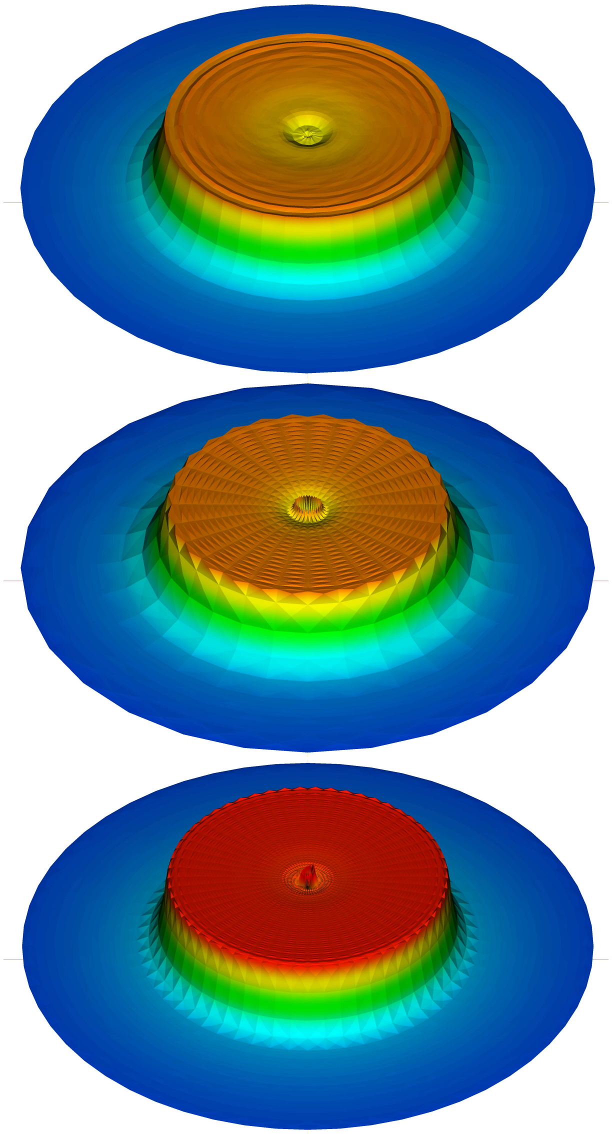

Figure 10.19. Noh test on a quadrant: Density elevation plots. Top: Mesh (a). Center: Mesh (b). Bottom: Mesh (c).

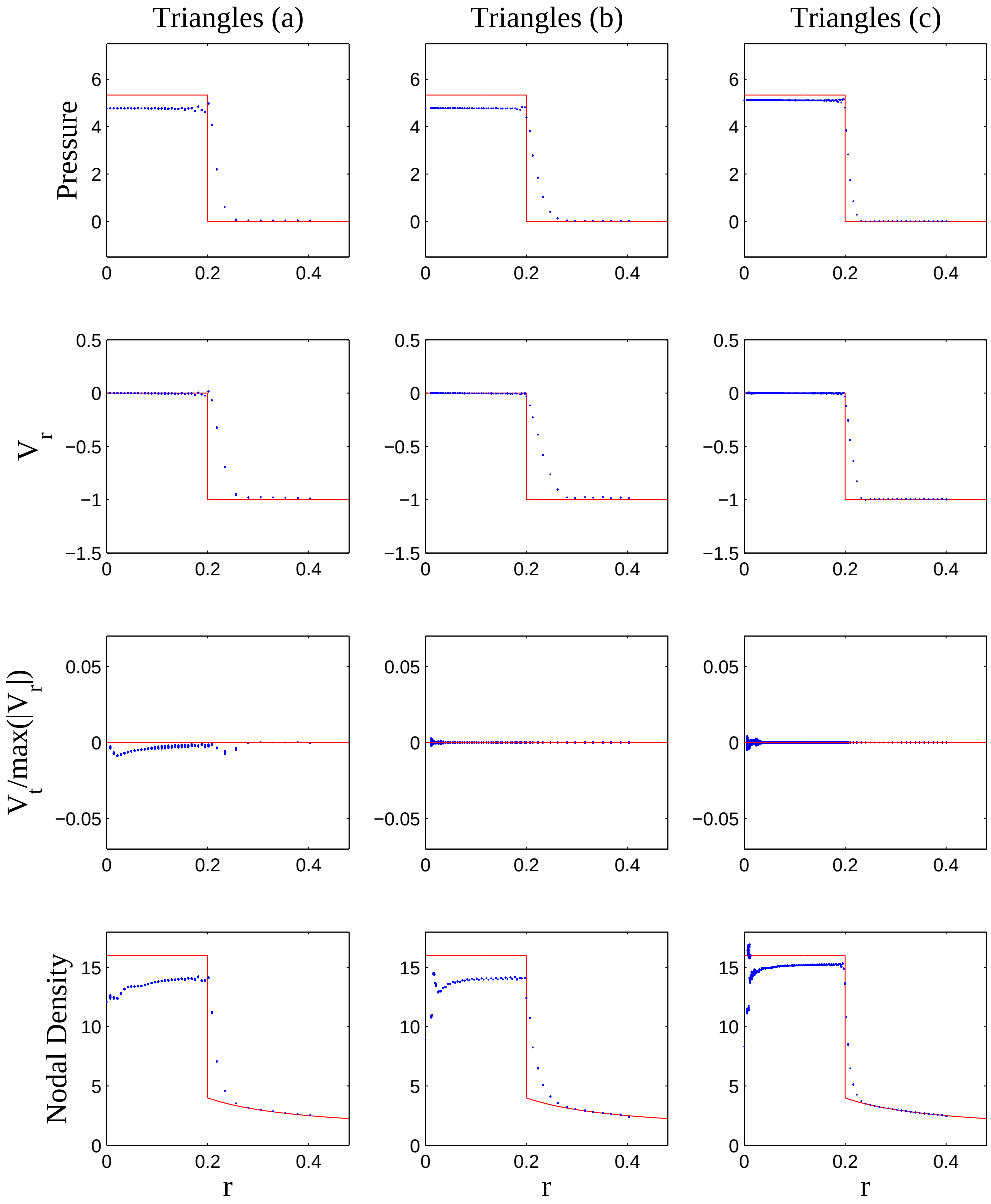

Figure 10.20. Noh test performance comparison. Left: Mesh (a). Center: Mesh (b). Right: Mesh (c). Along the rows from top to bottom are plotted – as a function of the radius $r = \sqrt{x^2 + y^2}$ – pressure, radial velocity v_r, normalized tangential velocity ($v_t/\max(|v_r|)$), and nodal density, respectively. The exact solution is represented by the continuous line, the dots represent the nodal values of the numerical solution. All plots contain information from each angularly-displaced radial line.

10.3.2 Orthogonal grids on the unit quadrant

The next variants are even more demanding, since they are performed on meshes that do not posses cylindrical symmetry. We considered three meshing strategies for a Noh test on the quadrant $[0,1] \times [0,1]$: a mesh of 50×50 square Q1 elements, indicated as mesh (I), and two P1 meshes obtained by splitting each of the square along one or the other diagonal (mesh (II) and mesh (III)). A view near the origin of the initial grids used is presented in Figure 10.21. Zero normal velocity ("roller") boundary conditions are imposed along the bottom and left boundaries, corresponding to the x- and y-axis. Such boundary conditions are not exactly the same as symmetry boundary conditions specified in [8, 7], but the results show good agreement. The simulations for this set of meshes were performed at $CFL = 0.9$.

Deformed grids and density color plots are presented in Figures 10.22– 10.24. A comparison of elevation plots for the density is presented in Figure 10.25, while quantitative comparisons are presented in Figure 10.26. The first observation to be made is that the mesh does not possess radial symmetry, but just an approximate (machine precision) axis of symmetry given by the bi-secant of the quadrant (the line from the origin at a 45°-degree angle). This symmetry can be best checked by looking at the normalized tangential velocity plots in Figure 10.26. In order to preserve symmetry with respect to the quadrant bisecant, the plot of the tangential velocity must be perfectly symmetric about the horizontal axis. Although hard to see by the unexperienced eye, symmetry is mildly broken for mesh (I). Notwithstanding this fact, the solution in terms of pressure, radial velocity and density is quite good. In particular, the density profile for the mesh (I) appears very smooth. The smallest error for the density is however observed on mesh (II), while mesh (III) tends to be noisier. There is an explanation for this fact: Element edges in mesh (II) are better aligned with the shock front than the edges of mesh (III), which possess the worst alignment.

Overall, the quality of the presented results for the Noh test and its variants on both quadrilateral and triangular elements is in good agreement with state-of-the-art computations on quadrilaterals [8, 7] (see Fig. 10.27).

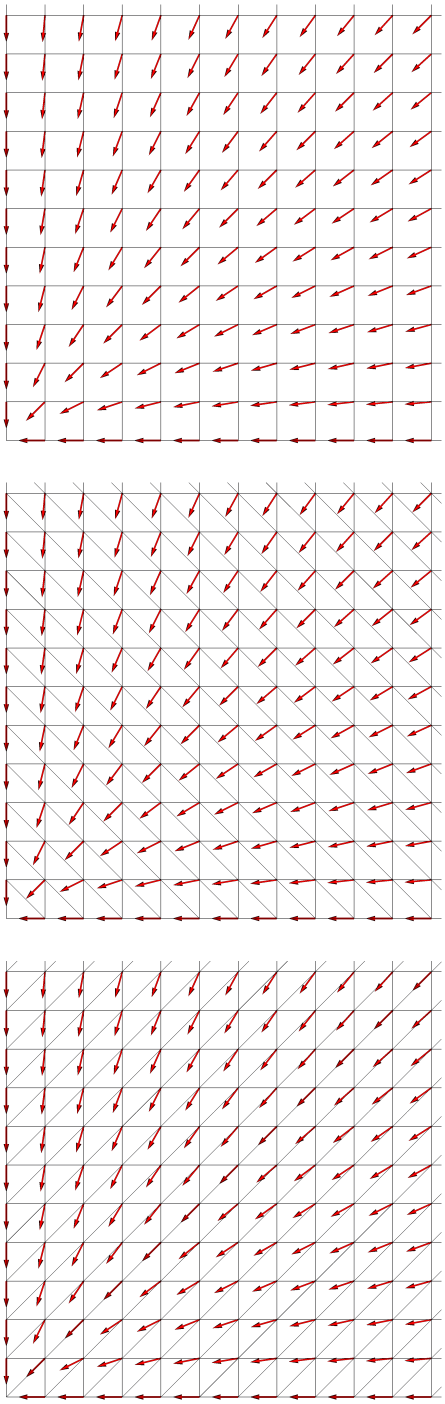

Figure 10.21. Initial meshes and velocity conditions for the Noh test on a quadrant. Top: Mesh (I). Center: Mesh (II). Bottom: Mesh (III).

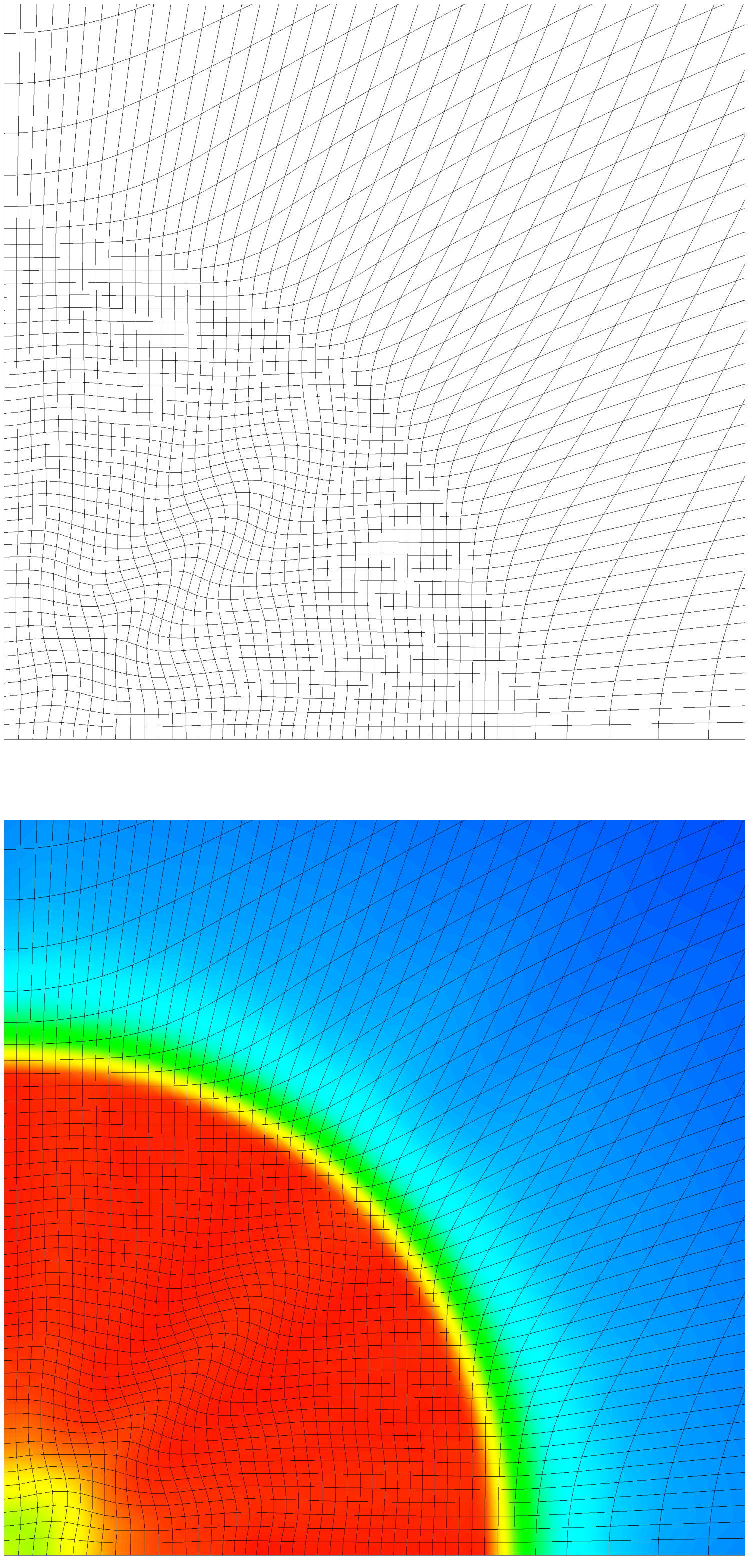

Figure 10.22. Noh test on a quadrant, mesh (I). Top: Deformed grid. Bottom: Contour plot of the density.

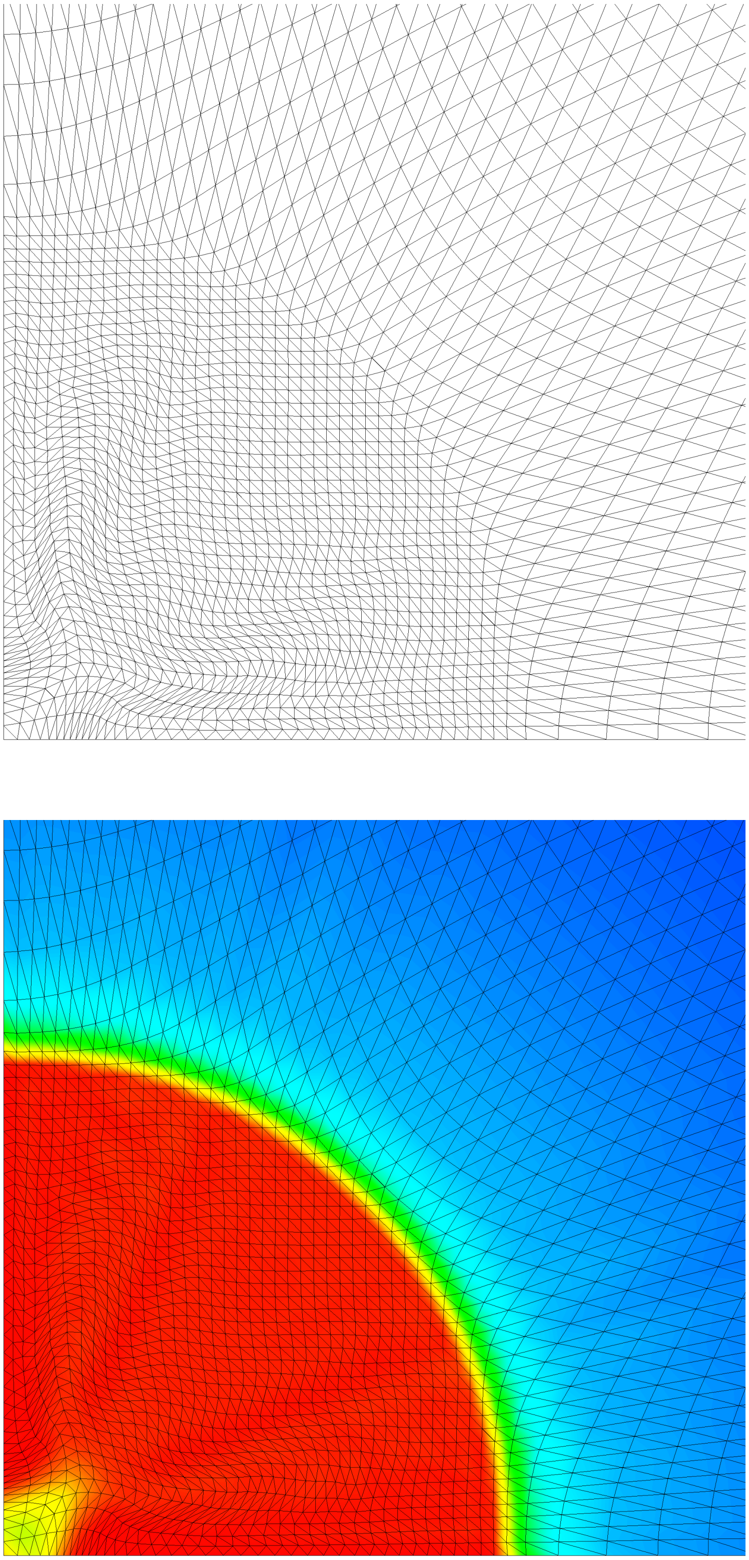

Figure 10.23. Noh test on a quadrant, mesh (II). Top: Deformed grid. Bottom: Contour plot of the density.

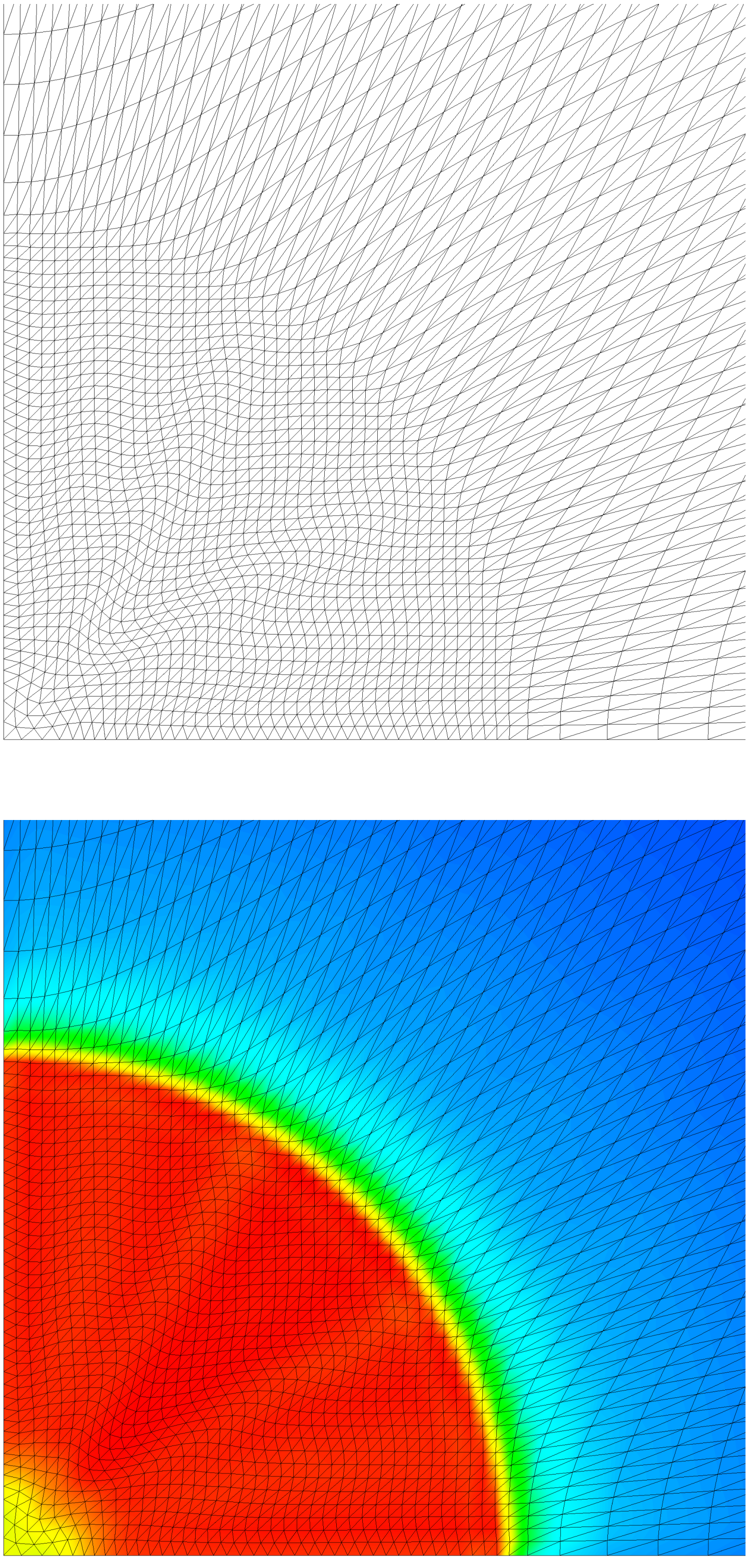

Figure 10.24. Noh test on a quadrant, mesh (III). Top: Deformed grid. Bottom: Contour plot of the density.

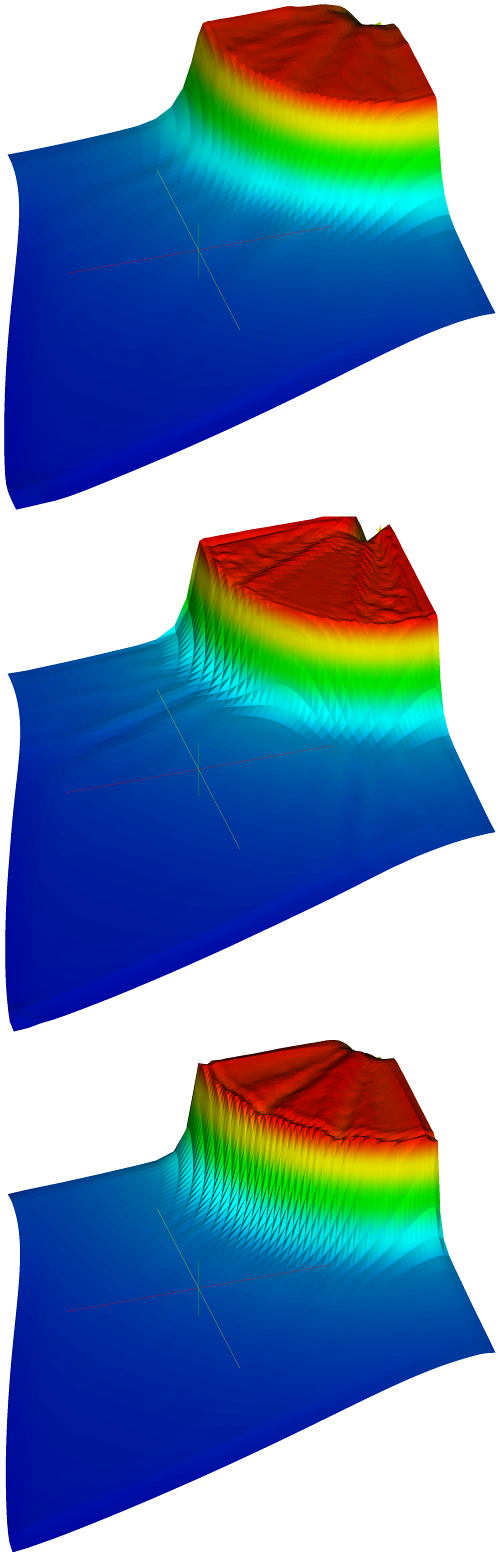

Figure 10.25. Noh test: Density elevation plots. Top: Mesh (I). Center: Mesh (II). Bottom: Mesh (III).

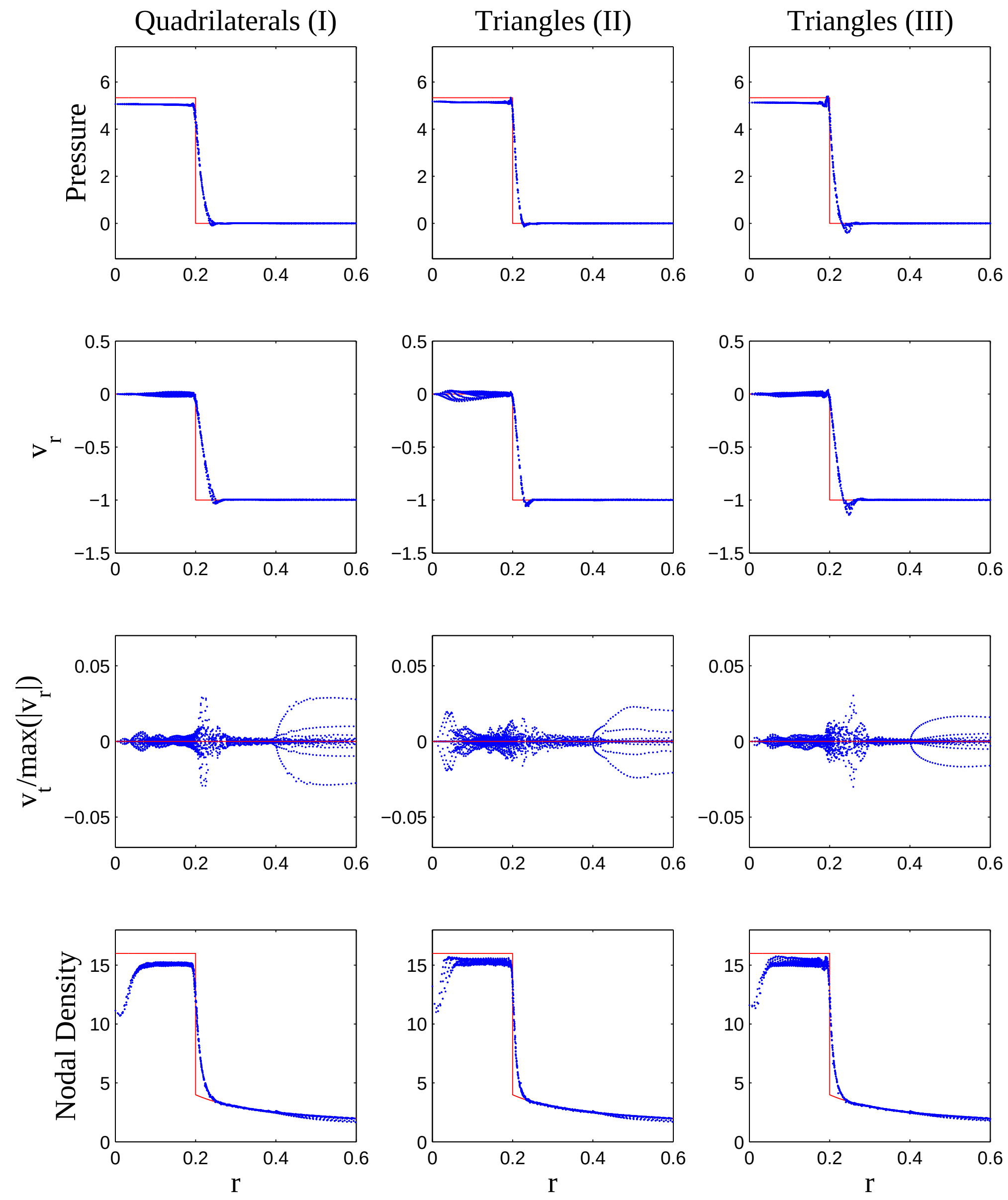

Figure 10.26. Noh test performance comparison. Left: Mesh (I). Center: Mesh (II). Right: Mesh (III). Along the rows from top to bottom are plotted – as a function of the radius $r = \sqrt{x^2 + y^2}$ – pressure, radial velocity v_r, normalized tangential velocity $(v_t/\max(|v_r|))$, and nodal density, respectively. The exact solution is represented by the continuous line, the dots represent the nodal values of the numerical solution. All plots contain information from each angularly-displaced radial line.

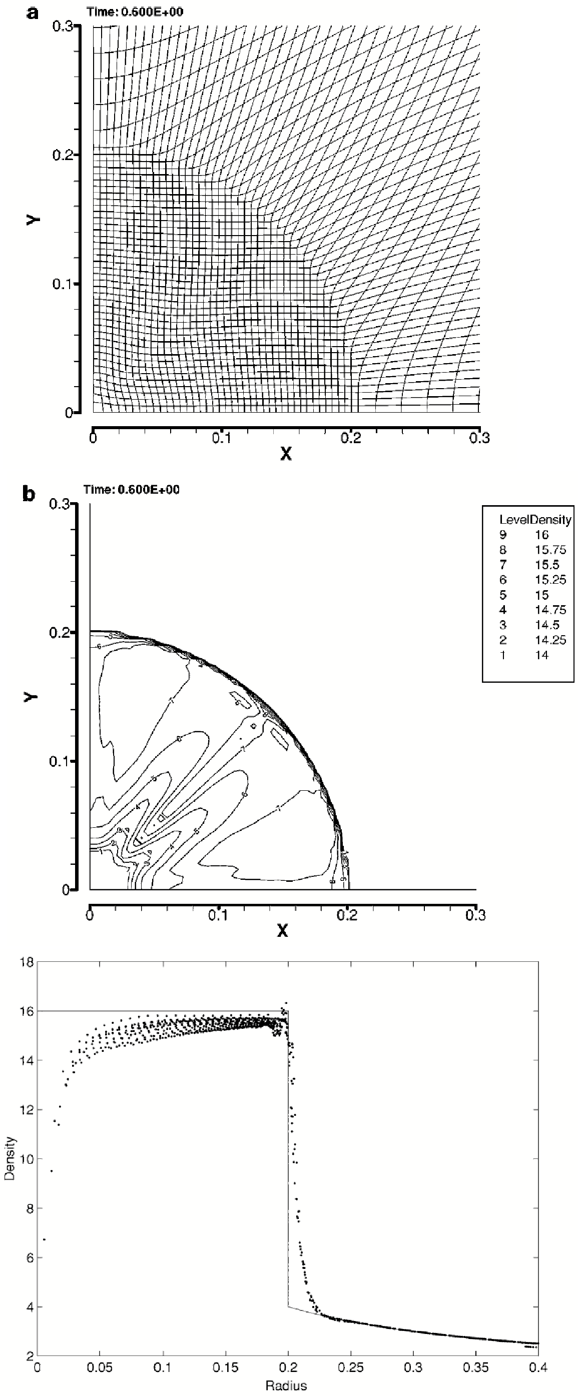

Figure 10.27. Results from [8] on the Noh test on the quadrant. The mesh is the same as the quadrilateral mesh (I). Above: Mesh deformation. Center: isolines for the density. Bottom: Density radial plot.

10.4 Sedov two-dimensional blast test

The proposed version of the Sedov blast test (an exact solution using self-similarity arguments can be found in [42]) assesses again the ability of the method to represent a cylindrical shock-wave pattern using a cartesian mesh.

A first set of three variants was computed on the $[0, 1.1] \times [0, 1.1]$ quadrant, with similar topology to the meshes (I), (II), and (III) adopted for the Noh test. We will keep the same nomenclature, indicating with mesh (I) a Q1 mesh of 45×45 squares, and with mesh (II) and (III), the P1 grids obtained by splitting each of the elements of mesh (I) in half along the diagonals. The initial mesh configurations, for the sake of brevity, are not shown, being very similar to the ones used in the Noh test.

The initial density has a uniform unit distribution, the energy is "zero" (actually 10^{-14}) everywhere, except the first square zone on the bottom left corner of the quadrant, near the origin, where it takes the value 409.7. Results in terms of the deformed grids and density contour plots are presented in Figures 10.28–10.30. Elevation plots of the nodal density are presented in Figure 10.31. Quantitative plots are pesented in Figure 10.32. The simulations for this set of meshes were performed at $CFL = 0.9$. The quality of the results is fairly high. Notice that the shock waves possess the correct arrival time, and the slight undershoot in the velocity for the triangular meshes. The density plots are in good agreement with the exact solution, and mesh (II) seems to deliver the best performance.

Additional qualitative plots for an anisotropic mesh of aspect ratio 1:2 are shown in Figure 10.33. From the contour plot of the density, the circular profile of the shock front is quite evident. Figure 10.34 shows results for an half-plane blast. It can be clearly seen that symmetry is preserved.

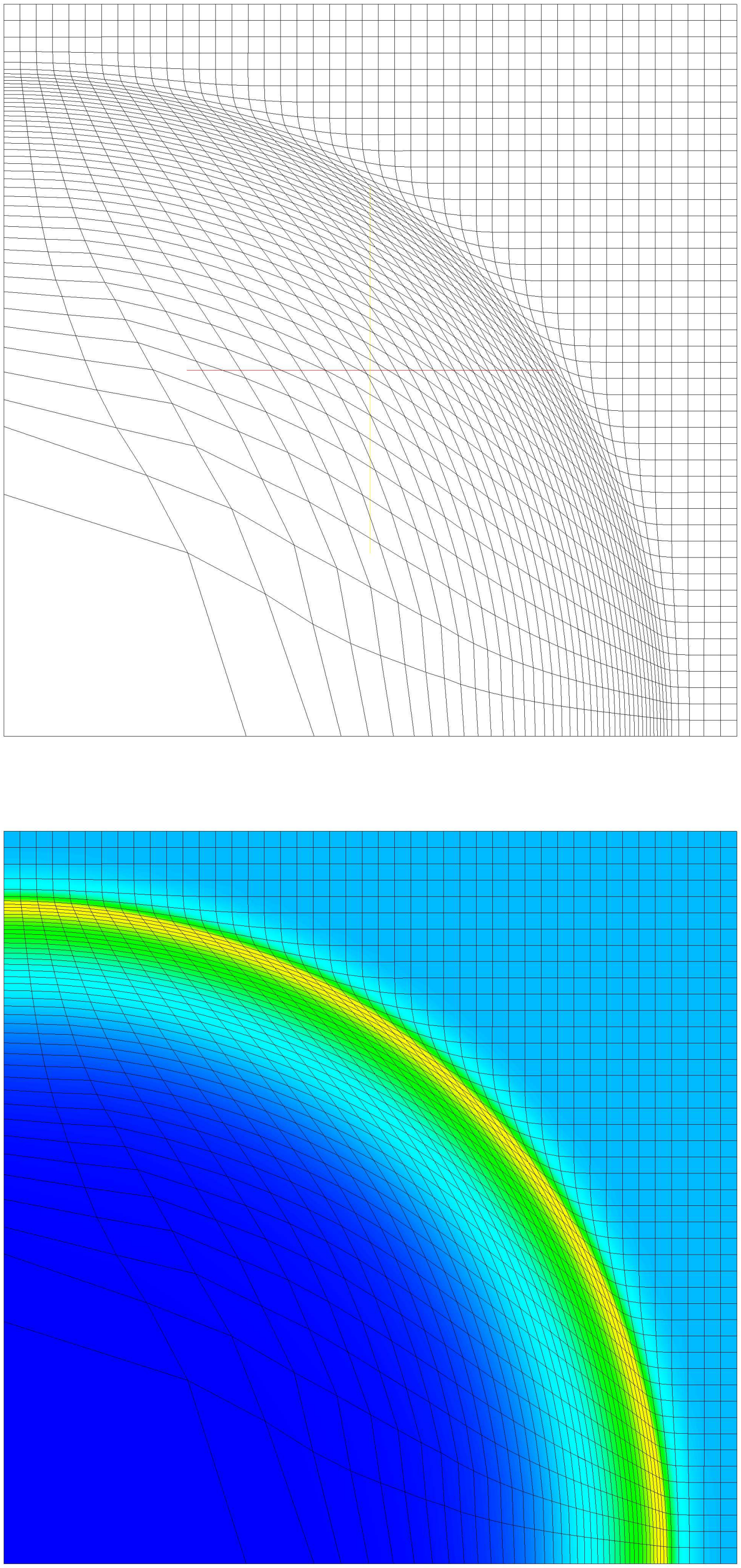

Figure 10.28. Sedov test on mesh (I). Top: Deformed grid. Bottom: Density contour plot.

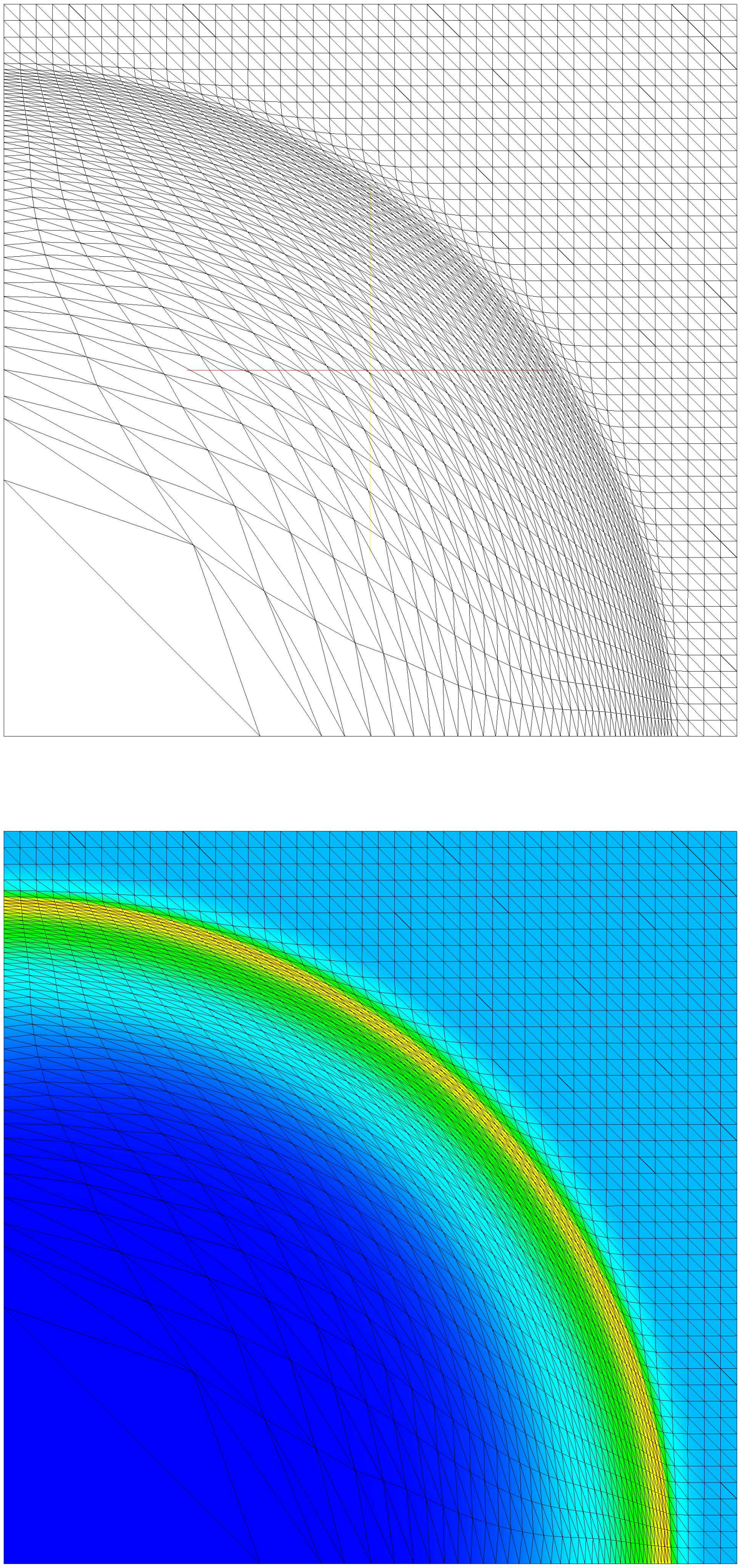

Figure 10.29. Sedov test on mesh (II). Top: Deformed grid. Bottom: Density contour plot.

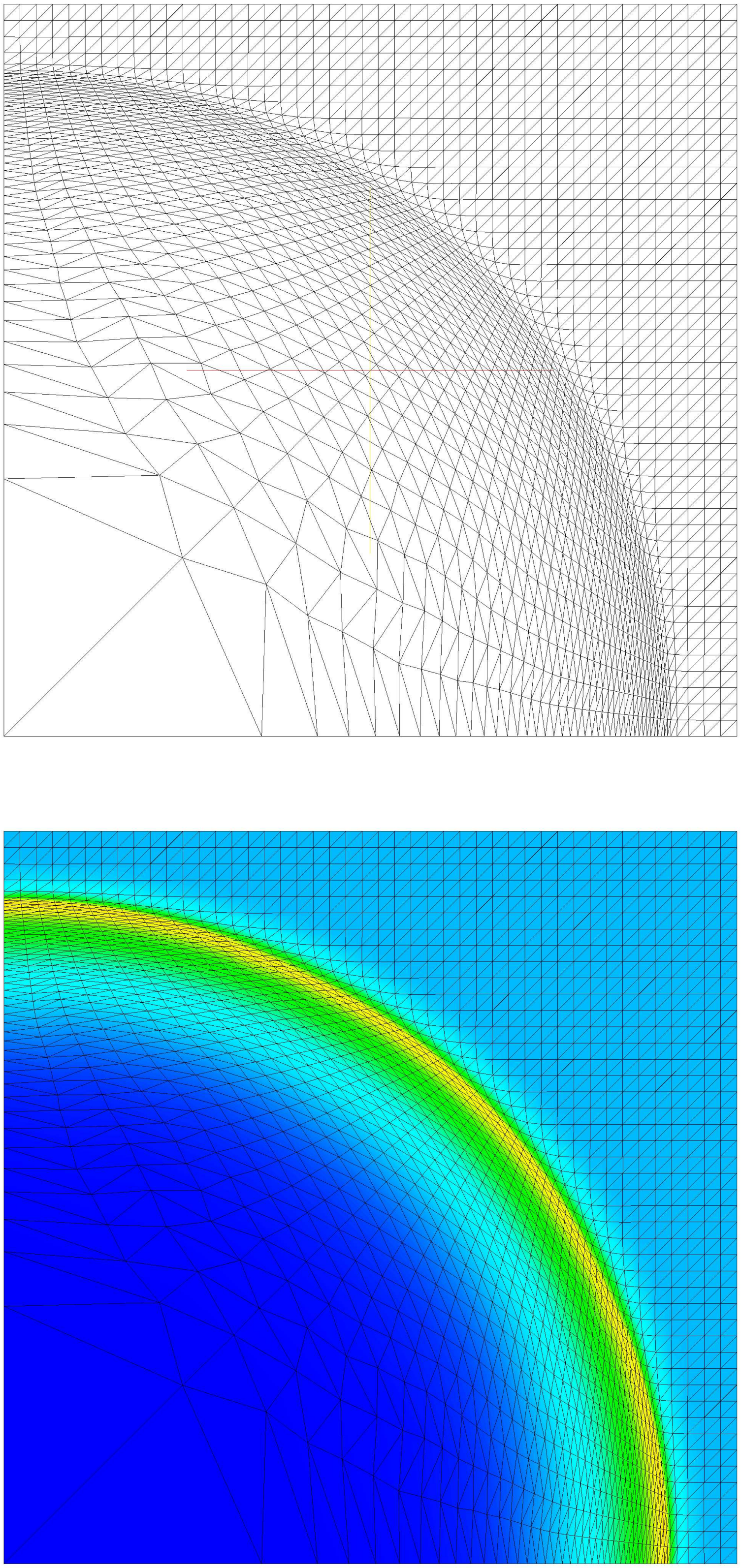

Figure 10.30. Sedov test on mesh (III). Top: Deformed grid. Bottom: Density contour plot.

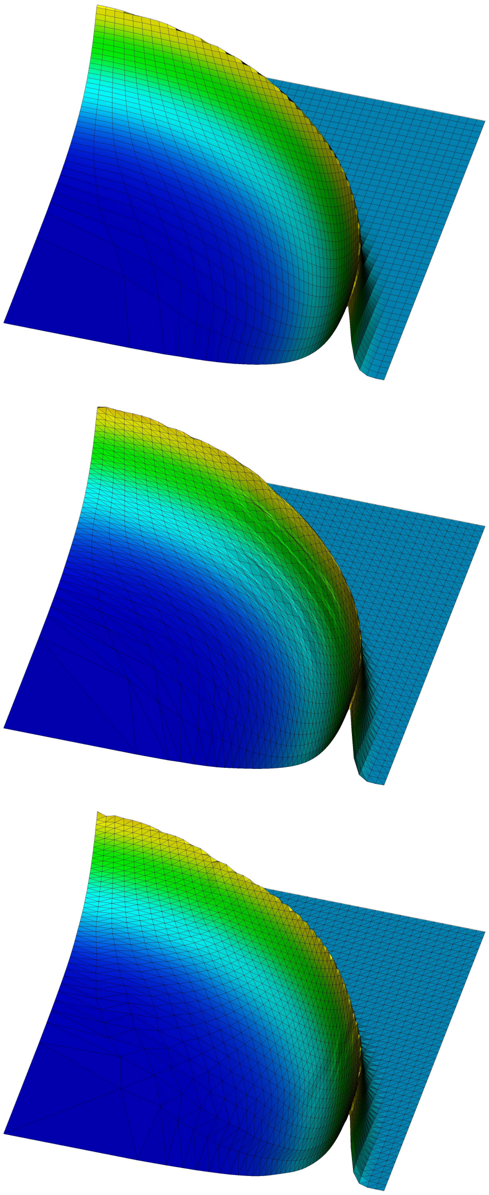

Figure 10.31. Sedov test: Density elevation plots. Top: Mesh (I). Center: Mesh (II). Bottom: Mesh (III).

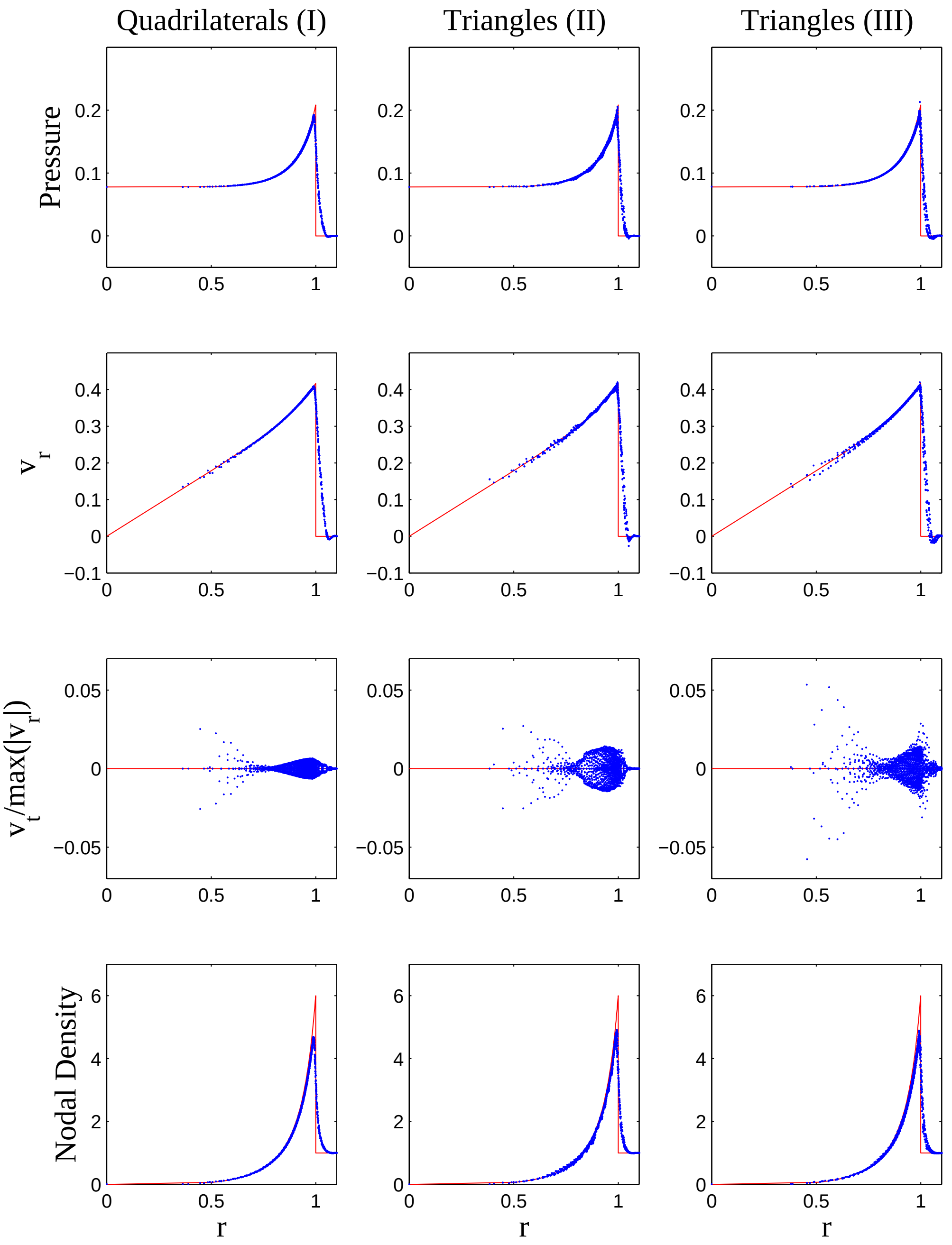

Figure 10.32. Sedov test performance comparison. Left: Mesh (I). Center: Mesh (II). Right: Mesh (III). Along the rows from top to bottom are plotted – as a function of the radius $r = \sqrt{x^2 + y^2}$ – pressure, radial velocity v_r, normalized tangential velocity $(v_t/\max(|v_r|))$, and nodal density, respectively. The exact solution is represented by the continuous red line, the dots represent the nodal values of the numerical solution. All plots contain information from each angularly-displaced radial line.

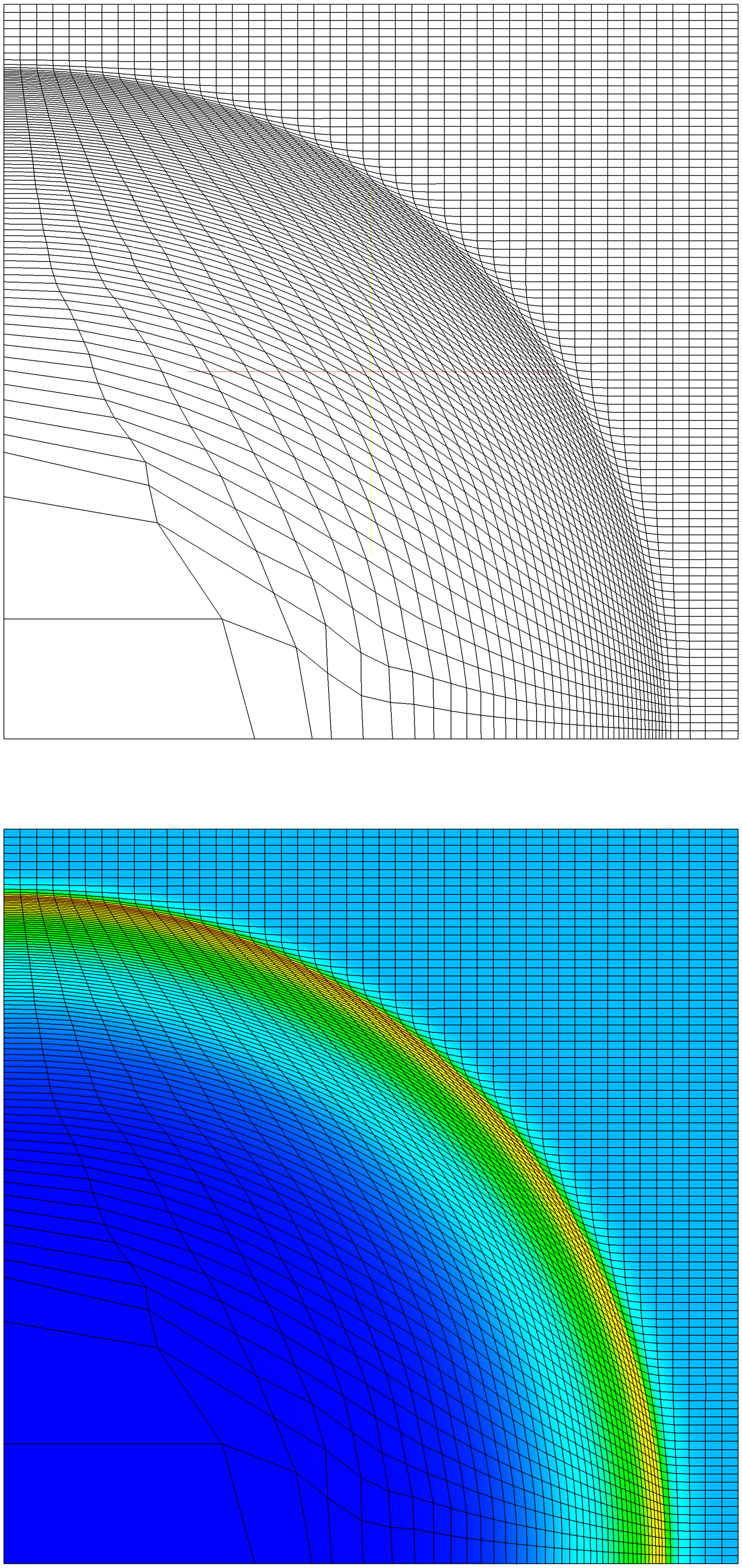

Figure 10.33. Sedov test on 1:2 aspect ratio anisotropic mesh. Top: Grid deformation. Bottom: Density contour plot.

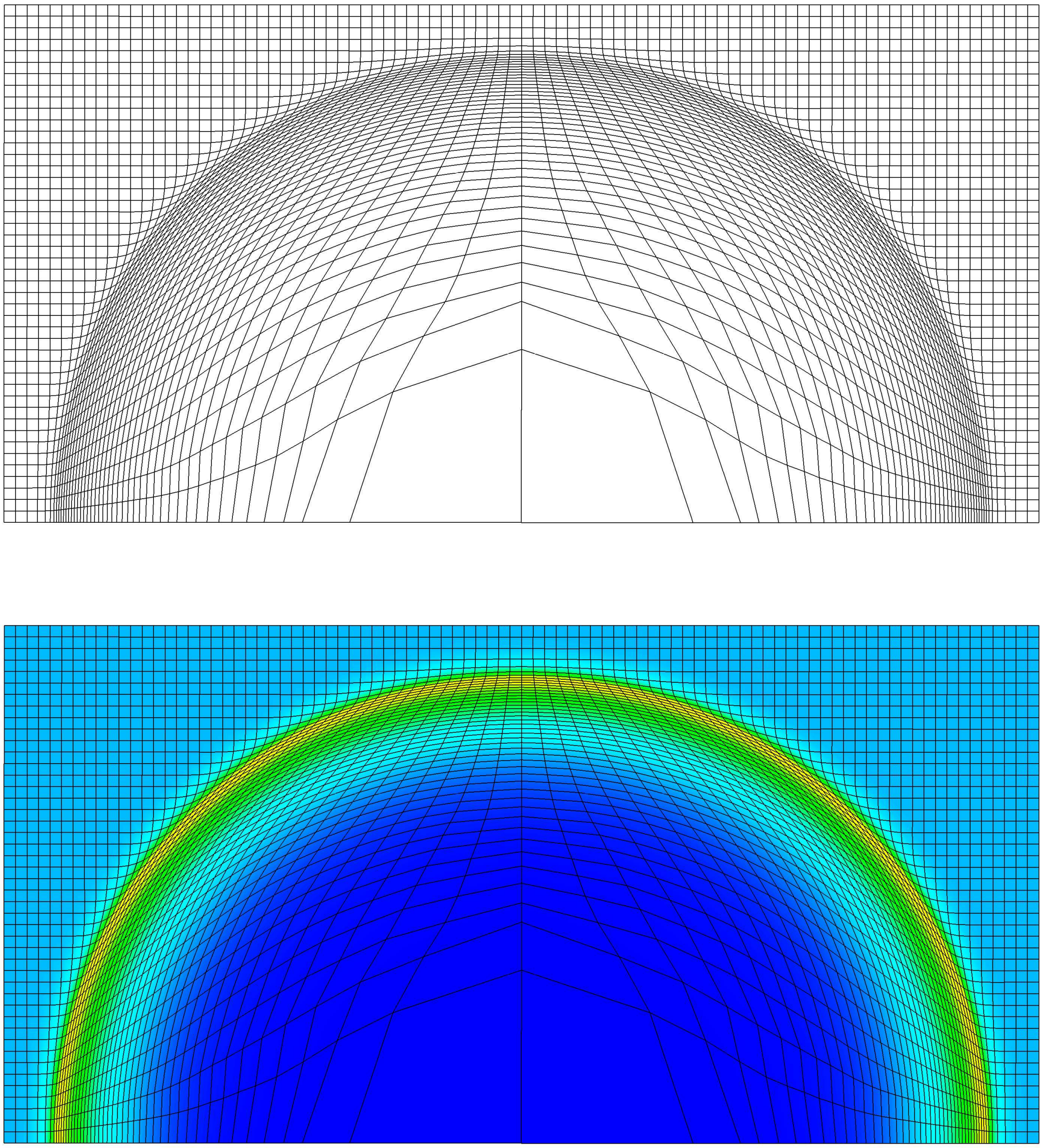

Figure 10.34. Sedov test on the half plane, quadrialteral mesh. Top: Grid deformation. Bottom: Density contour plot.

Chapter 11

Summary

A new SUPG approach to Lagrangian hydrodynamics has been proposed and described in detail. The main features of the method are:

1. A very natural representation of the gradients of thermodynamic variables, with very positive consequences in all test cases in which geometrical symmetries of the solution do not match the geometrical structure of the mesh.

2. The method addresses the issue of appropriate pressure gradient representation from a different perspective with respect to mimetic/compatible finite difference discretizations [8]. A compatible discretization uses the standard, cell-centered, piece-wise constant approximation for the pressure, and develops a consistent way of representing its gradient on the edges of the cells. The SUPG approach developed herein adopts isoparametric elements, for which there is a natural description of gradient operators in the interiors of the elements.

3. The formulation has been developed for the fairly large class of materials obeying Mie-Grüneisen equations of state.

Numerical results have indicated that a natural representation of the pressure gradients obtained with isoparametric finite element has very positive consequences on the overall quality of solutions.

The method has been proven to be reliable and accurate not only for isoparametric quadrilateral elements, but also for triangular P1 elements, and can be extended, if needed, to higher-order elements.

Three-dimensional extensions, although more involved in their implementation, are not foreseen to present any additional theoretical issues, and are currently under development.

Additional work is needed to improve the artificial viscosity operator, and to address the issue of positivity preservation for pressure. However, the quality of the results is already very encouraging, and demonstrates the potential of the method.

The capability of combining simplex and brick elements on the same computational mesh is under development. The goal is attain greater flexibility in mesh generation.

Further developments are anticipated in the direction of an Arbitrary Lagrangian Eulerian (ALE) formulation, and more complex constitutive laws. Additional research in the case of materials with strength for structural mechanics applications is also currently envisioned.

Appendix A

An introduction to SUPG stabilization

The reader familiar with classical Lagrangian hydrocode technology, but unfamiliar with the SUPG approach, may have some difficulty in grasping the key aspects of the formulation presented herein. It is therefore felt useful to give a brief review of the method in the simple context of the steady scalar advection-diffusion problem in one dimension

$$c\partial_x\phi - \kappa\partial_{xx}\phi = f, \text{ on } [0,1] \tag{A.1}$$

with Dirichlet boundary conditions applied at the ends of the interval $[0,1]$. Let us consider the Galerkin formulation which makes use of piecewise linear, globally continuous basis functions with local support, and reads

$$-\int_0^1 c\partial_x w^h \phi^h dx + \int_0^1 \kappa\partial_x w^h \partial_x \phi^h dx - \int_0^1 w^h f dx = 0 \tag{A.2}$$

The boundary conditions are embedded in the function space representing ϕ^h and w^h is assumed to vanish on the boundary. On a uniform partition $[0,1] = \bigcup_{e=1}^{n_{el}}[x_e, x_{e+1}]$, the assembly of the advective first derivative integrals results in the central difference stencil:

$$\mathop{\mathsf{A}}_{e=A}^{A+1}\left(-\int_{x_{e-1}}^{x_e} \partial_x w^h c\phi^h dx\right) = c\;\Delta x\;\frac{\phi^h_{A+1} - \phi^h_{A-1}}{2\Delta x} \tag{A.3}$$

where A is the assembly operator [2, 18], and ϕ^h_{A-1} and ϕ^h_{A+1} are nodal degrees-of-freedom of the approximate solution ϕ^h. Since the central difference scheme is prone to node-to-node oscillations near sharp layers of the solution, a standard Galerkin formulation is not robust in the advection-dominated case (i.e., $|c| \to \infty$). A more robust approach would be to use an upwind difference,

$$c\frac{\partial\phi}{\partial x} \approx H(c)c\frac{\phi^h_A - \phi^h_{A-1}}{\Delta x} + (1 - H(c))c\frac{\phi^h_{A+1} - \phi^h_A}{\Delta x} \tag{A.4}$$

where $H(c)$ is the Heaviside function, equal to one if $c > 0$ and zero otherwise. Unfortunately, a simple-minded incorporation of the upwinding concept into a Galerkin formulation would lead to lack of consistency and, ultimately, lack of optimal convergence rates. One of the key innovations proposed with the SUPG stabilization method was to incorporate upwinding in a *variationally consistent* way, leading to optimal convergence rates in the entire range of choices for the advection-diffusion parameters c and κ.

The starting point of the SUPG method is the observation that an upwind discretization can be obtained by perturbing the central difference stencil using an appropriate numerical viscosity:

$$\begin{aligned} & H(c)c\frac{\phi_A^h - \phi_{A-1}^h}{\Delta x} + (1 - H(c))c\frac{\phi_{A+1}^h - \phi_A^h}{\Delta x} \\ & \qquad = c\frac{\phi_{A+1}^h - \phi_{A-1}^h}{2\Delta x} + \frac{|c|\Delta x}{2}\ \frac{\phi_{A+1}^h - 2\phi_A^h + \phi_{A-1}^h}{\Delta x^2} \end{aligned} \tag{A.5}$$

A variationally consistent way of recovering (A.5) is given by the SUPG method:

$$\begin{aligned} & -\int_0^1 c\partial_x w^h \phi^h dx + \int_0^1 \kappa \partial_x w^h \partial_x \phi^h dx - \int_0^1 f w^h dx \\ & \qquad -\sum_{e=1}^{n_{el}} \int_{x_{e-1}}^{x_e} \mathcal{L}_{adv}^* w^h \tau Res(\phi^h) dx \;=\; 0 \end{aligned} \tag{A.6}$$

where $\mathcal{L}_{adv}^* = -c\partial_x$ is the adjoint of the advective part of the differential operator, and $Res(\phi^h) = c\partial_x\phi^h + \kappa\partial_{xx}\phi^h - f$ is the strong *residual.* The scalar parameter τ is a function of the element Péclet number $Pe_h = c\Delta x/(2\kappa)$, and yields a *nodally exact* solution for piecewise-constant f, for *all* values of Pe_h:

$$\tau = \frac{\Delta x}{2|c|}\left(\coth Pe_h - \frac{1}{Pe_h}\right) \tag{A.7}$$

This result holds on a non-uniform mesh with τ defined element-wise [23].

Note that

$$\lim_{Pe_h\to\infty}\left(\coth Pe_h - \frac{1}{Pe_h}\right) = 1 \tag{A.8}$$

$$\lim_{Pe_h\to 0}\left(\coth Pe_h - \frac{1}{Pe_h}\right) = \frac{|c|\Delta x}{6\kappa} \tag{A.9}$$

In the advection-dominated case, of particular interest in Lagrangian hydrodynamics, the two-element assembly of the term

$$\begin{aligned}\mathop{\mathsf{A}}_{e=A}^{A+1}\int_{x_{e-1}}^{x_e} -\mathcal{L}^*_{adv} w^h \tau c \partial_x \phi^h dx &= \mathop{\mathsf{A}}_{e=A}^{A+1}\int_{x_{e-1}}^{x_e} \frac{\Delta x|c|}{2}\,\partial_x w^h\,\partial_x \phi^h \\ &= \Delta x \frac{|c|\Delta x}{2}\,\frac{\phi^h_{A+1}-2\phi^h_A+\phi^h_{A-1}}{\Delta x^2} \qquad \text{(A.10)}\end{aligned}$$

yields the numerical viscosity in (A.5). The full stabilization operator also includes a term depending on the force f:

$$\mathop{\mathsf{A}}_{e=A}^{A+1}\int_{x_{e-1}}^{x_e} -\mathcal{L}^*_{adv} w^h \tau f dx = \mathop{\mathsf{A}}_{e=A}^{A+1}\int_{x_{e-1}}^{x_e} \frac{\Delta x|c|}{2}\,\partial_x w^h\ f \qquad \text{(A.11)}$$

Due to (A.11), the SUPG approach retains a residual-consistent structure, and is clearly different from plain upwinding.

Remark 25 *Consistency is due to the* residual structure *of the stabilization operator.*

Remark 26 *Effectively,* $-\tau\mathcal{L}^*_{adv}w^h$ *is a* perturbation *to the test function, from which the name Streamline-Upwind Petrov-Galerkin (SUPG) formulation derived.*

Remark 27 *It can be proved that the SUPG method entails optimal convergence rates for all values of the local Péclet number* Pe_h*. In multiple dimensions it is not possible to recover a nodally exact approximation to the solution, but optimal error estimates still hold.*

Remark 28 *It is very easy to generalize the formulation to unsteady flows, when a space-time formulation is adopted.*

Appendix B

One-dimensional stabilization [44]

In the following derivations, $\boldsymbol{A}_0$ and $\boldsymbol{A}_i$ refer only to the momentum and energy blocks of the same matrices, since stabilization is not applied to the ODE relating rate of displacements to velocities. The multi-dimensional definition given by Shakib, Hughes and Johan [44], a fairly standard definition of the $\boldsymbol{\tau}$ matrix, reads:

$$\boldsymbol{\tau} = \boldsymbol{A}_0^{-1}\left(\boldsymbol{C}^2 + \left(\frac{\partial \xi_0}{\partial t}\right)^2 \boldsymbol{I} + \frac{\partial \xi_i}{\partial X_j}\frac{\partial \xi_i}{\partial X_k}\hat{\boldsymbol{A}}_j\hat{\boldsymbol{A}}_k\right)^{-1/2} \tag{B.1}$$

where $\hat{\boldsymbol{A}}_1 = \boldsymbol{A}_1\boldsymbol{A}_0^{-1}$ and ξ_i are the coordinates in the parent domain of each element, and ξ_0 refers to the time axis. For an ideal gas in one dimension:

$$\frac{\partial \xi_i}{\partial X_j}\frac{\partial \xi_i}{\partial X_k}\hat{\boldsymbol{A}}_j\hat{\boldsymbol{A}}_k = \left(\frac{2}{\Delta X}\right)^2 \hat{\boldsymbol{A}}_1{}^2 \tag{B.2}$$

Therefore:

$$\begin{aligned}\hat{\boldsymbol{A}}_1 = \boldsymbol{A}_1\boldsymbol{A}_0^{-1} &= \begin{bmatrix} 0 & 1 \\ \frac{\gamma}{\gamma-1}p & 0 \end{bmatrix}\begin{bmatrix} \rho_0 & 0 \\ 0 & \frac{J}{\gamma-1} \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 \\ \frac{\gamma}{\gamma-1}p & 0 \end{bmatrix}\begin{bmatrix} \frac{1}{\rho_0} & 0 \\ 0 & \frac{\gamma-1}{J} \end{bmatrix} \\ &= \begin{bmatrix} 0 & \frac{\gamma-1}{J} \\ \frac{\gamma}{\gamma-1}\frac{p}{\rho_0} & 0 \end{bmatrix}\end{aligned} \tag{B.3}$$

$$\hat{\boldsymbol{A}}_1{}^2 = \begin{bmatrix} \frac{\gamma p}{\rho_0 J} & 0 \\ 0 & \frac{\gamma p}{\rho_0 J} \end{bmatrix} = \left(\frac{c_s}{J}\right)^2 \boldsymbol{I}_{2\times 2} \tag{B.4}$$

with $c_s = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\frac{\gamma p J}{\rho_0}}$. It is important to realize that the form of the SUPG stabilization is dependent on the function spaces adopted, and in particular on the time-integration strategy.

B.1 Second-order time integrator

For the second-order time integrator developed here, $\frac{\partial \xi_0}{\partial t} = \frac{2}{\Delta t}$ and

$$\begin{aligned}
\boldsymbol{\tau} &= \hat{\boldsymbol{A}}_0^{-1} \left(\left(\frac{2}{\Delta t} \right)^2 \boldsymbol{I}_{2\times 2} + \left(\frac{2\ c_s}{J\ \Delta X} \right)^2 \boldsymbol{I}_{2\times 2} \right)^{-1/2} \\
&= \frac{\Delta t/2}{\sqrt{1+\alpha^2}} \hat{\boldsymbol{A}}_0^{-1} \\
&= \frac{\Delta t}{2\sqrt{1+\alpha^2}} \begin{bmatrix} \frac{1}{\rho_0} & 0 \\ 0 & \frac{\gamma-1}{J} \end{bmatrix} \qquad \text{(B.5)}
\end{aligned}$$

where $\alpha = \frac{c_s \Delta t}{J \Delta X}$ is the local Courant number.

References

[1] A. K. Aziz and P. Monk. Continuous finite elements in space and time for the heat equation. *Mathematics of Computation*, **144**:70–97, 1998.

[2] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures.* J. Wiley & Sons, New York, 2000.

[3] D. J. Benson. A new two-dimensional flux-limited shock viscosity for impact calculations. *Computer Methods in Applied Mechanics and Engineering*, **93**:39–95, 1991.

[4] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, **99**:235–394, 1992.

[5] P. B. Bochev, M. D. Gunzburger, and J. N. Shadid. Stability of the SUPG finite element method for transient advection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, **193**:2301–2323, 2004.

[6] A. N. Brooks and T. J. R. Hughes. Streamline upwind / Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, **32**:199–259, 1982.

[7] J. C. Campbell and M. J. Shashkov. A tensor artificial viscosity using a mimetic finite difference algorithm. Technical report, LA-UR-00-2290, Los Alamos National Laboratory, 2000.

[8] J. C. Campbell and M. J. Shashkov. A tensor artificial viscosity using a mimetic finite difference algorithm. *Journal of Computational Physics*, **172**:739–765, 2001.

[9] E. J. Caramana, M. J. Shashkov, and P. P. Whalen. Formulations of artificial viscosity for multi-dimensional shock wave computations. *Journal of Computational Physics*, **144**:70–97, 1998.

[10] M. A. Christon. Pressure gradient approximations for staggered-grid finite element hydrodynamics. *International Journal of Numerical Methods in Fluids*, 2005. In preparation.

[11] W. Dettmer and D. Perić. An analysis of the time integration algorithms for the finite element solutions of incompressible Navier-Stokes equations based on a stabilised formulation. *Computer Methods in Applied Mechanics and Engineering*, **192**(9-10):1177–1226, 2003.

[12] D. Estep and D. A. French. Global error control for the continuous Galerkin finite element method for ordinary differential equations. *RAIRO*, **28**:815–852, 1994.

[13] D. A. French. A space-time finite element method for the wave equation. *Computer Methods in Applied Mechanics and Engineering*, **107**:145–157, 1993.

[14] D. A. French. Continuous Galerkin finite element methods for a forward-backward heat equation. *Numerical Methods for Partial Differential Equations*, **15**:491–506, 1999.

[15] D. A. French and S. Jensen. Long time behaviour of arbitrary order continuous time Galerkin schemes for some one-dimensional phase transition problems. *IMA Journal on Numerical Analysis*, **14**:421–442, 1994.

[16] D. A. French and T. E. Peterson. A continuous space-time finite element method for the wave equation. *Mathematics of Computation*, **65**:491–506, 1996.

[17] G. Hauke and T. J. R. Hughes. A comparative study of different sets of variables for solving compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, **153**:1–44, 1998.

[18] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1987. (Dover reprint, 2000).

[19] T. J. R. Hughes, G. Engel, L. Mazzei, and M. Larson. The continuous Galerkin method is locally conservative. *Journal of Computational Physics*, **163**(2):467–488, 2000.

[20] T. J. R. Hughes, L. P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, **73**:173–189, 1989.

[21] T. J. R. Hughes, L. P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, **63**:97–112, 1987.

[22] T. J. R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, **58**:305–328, 1986.

[23] T.J.R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid-scale models, bubbles and the origin of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, **127**:387–401, 1995.

[24] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J.-B. Quincy. The Variational Multiscale Method — a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, **166**:3–24, 1998.

[25] T.J.R. Hughes and T. E. Tezduyar. Finite element methods for first-order hyperbolic system with particular emphasis on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering*, **45**:217–284, 1984.

[26] B. L. Hulme. Discrete Galerkin and related one-step methods for ordinary differential equations. *Mathematics of Computation*, **26**-120:881–891, 1972.

[27] P. Jamet. Stability and convergence of a generalized Crank-Nicolson scheme on a variable mesh for the heat equation. *SIAM Journal on Numerical Analysis*, **17**(4):530–539, 1980.

[28] C. Johnson, U. Nävert, and J. Pitkäranta. Finite element methods for linear hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, **45**:285–312, 1984.

[29] C. Johnson and A. Szepessy. On the convergence of a finite element method for a nonlinear hyperbolic conservation law. *Mathematics of Computation*, **49**:427–444, 1987.

[30] C. Johnson, A. Szepessy, and P. Hasnbo. On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws. *Mathematics of Computation*, **54**:107–129, 1990.

[31] O. Klaas, A. Maniatty, and M.S. Shephard. A stabilized mixed finite element method for finite elasticity. Formulation for linear displacement and pressure interpolation. *Computer Methods in Applied Mechanics and Engineering*, **180**:65–79, 1999.

[32] V. F. Kuropatenko. On difference methods for the equations of hydrodynamics. In N. N. Janenko, editor, *Difference methods for solutions of problems of mathematical physics, I*. American Mathematical Society, Providence, RI, 1967.

[33] R. Loubère. Investigation of triangular meshes for compressible Lagrangian hydrodynamics. Technical report, LA-UR-05-2937, Los Alamos National Laboratory, May 2005.

[34] A. Maniatty and Y. Liu. Stabilized finite element method for viscoplastic flow: formulation with state variable evolution. *International Journal of Numerical Methods in Engineering*, **56**:185–209, 2003.

[35] A. Maniatty, Y. Liu, O. Klaas, and M.S. Shephard. Stabilized finite element method for viscoplastic flow: formulation and a simple progressive solution strategy. *Computer Methods in Applied Mechanics and Engineering*, **190**:4609–4625, 2001.

[36] A. Maniatty, Y. Liu, O. Klaas, and M.S. Shephard. Higher order stabilized finite element method for hyperelastic finite deformation. *Computer Methods in Applied Mechanics and Engineering*, **191**:1491–1503, 2002.

[37] L. G. Margolin. A centered artificial viscosity for cells with large aspect ratios. Technical report, UCRL-53882, Lawrence Livermore National Laboratory, 1988.

[38] J. E. Marsden and T. J. R. Hughes. *Mathematical Foundations of Elasticity*. Dover, Mineola, NY, 1994.

[39] W. F. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics*, **72**:78–120, 1987.

[40] G. Scovazzi. Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations. *Computer Methods in Applied Mechanics and Engineering*, 2005. Submitted.

[41] G. Scovazzi. *Multiscale methods in science and engineering*. PhD thesis, Mechanical Engineering Department, Stanford Universty, September 2004. Available at `http://www.cs.sandia.gov/~gscovaz/pubs.html`.

[42] L. I. Sedov. *Similarity and Dimensional Methods in Mechanics*. Academic Press, New York, 1959.

[43] F. Shakib and T. J. R. Hughes. A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms. *Computer Methods in Applied Mechanics and Engineering*, **87**:35–58, 1991.

[44] F. Shakib, T. J. R. Hughes, and Z. Johan. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, **89**:141–219, 1991.

[45] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, **27**:1–31, 1978.

[46] A. Szepessy. Convergence of a shock-capturing streamline diffusion finite element method for a scalar conservation law in two space dimensions. *Mathematics of Computation*, **53**:527–545, 1989.

[47] T. E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal of Numerical Methods in Fluids*, **43**:555–575, 2003.

[48] T. E. Tezduyar. Determination of the stabilization and shock-capturing parameters in SUPG formulation of compressible flows. In *Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004*, Jyäskylä, Finland, July 24-28, 2004.

[49] T. E. Tezduyar and M. Senga. Determination of the stabilization and shock-capturing parameters in SUPG formulation of compressible flows. In *Proceedings of WCCM VI in conjunction with APCOM04*, Beijing, China, September 5-10, 2004. Tsinghua University Press & Springer-Verlag.

[50] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction.* Springer-Verlag, Berlin-Heidelberg, 2nd edition, 1999.

[51] J. Von Neumann and R.D. Richtmyer. A method for the numerical computation of hydrodynamic shocks. *Journal of Applied Physics*, **21**:232–237, 1950.

[52] M. L. Wilkins. Use of artificial viscosity in multidimensional shock wave problems. *Journal of Computational Physics*, **36**:281–303, 1979.

[53] P. R. Woodward. Trade-offs in designing explicit hydrodynamics schemes for vector computers. In G. Rodrigue, editor, *Parellel Computation.* Academic Press, New York, 1982.

[54] P. R. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, **54**:115–173, 1984.